File Number S360-30
Form No. C30-5001-5

DOS

IBM

Systems Reference Library

IBM System/360
Disk Operating System
Basic Telecommunications Access Method

Program Number 360N-CQ-469

This publication describes the Basic Telecommunications Access Method (BTAM) used with the System/360 Disk Operating System (DOS) control program. BTAM provides READ/WRITE level macro instructions for the assembler-language programmer who is implementing programs for telecommunications applications.

Included are macro instruction formats and descriptions and specific information on device-dependent considerations.

For titles and abstracts of associated publications see the IBM System/360 Bibliography, Form A22-6822.

Sixth Edition, April 1968

This edition, C30-5001-5, is a major revision of, and obsoletes, C30-5001-4. In addition, it incorporates changes released in Technical Newsletters N30-5016, N30-5024, N30-5505, N30-5507, and N30-5508. With the exception of the support described for planning purposes (see the Note at the end of the Preface), the support described herein corresponds to Release 17 of the Disk Operating System (DOS). Changes to the text are indicated by a vertical line to the left of the change; revised illustrations are denoted by the symbol (●) to the left of the caption.

Significant changes or additions to the specifications contained in this publication are continually being made. When using this publication in connection with the operation of IBM equipment, check the latest SRL Newsletter for revisions or contact the local IBM branch office.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form is provided at the back of this publication for readers' comments. If the form has been removed comments may be addressed to IBM Corporation, Programming Documentation, Department 844, P.O. Box 12275, Research Triangle Park, North Carolina 27709.

Supported Devices

BTAM supports the following devices on nonswitched lines attached to a multiplexor channel through the IBM 2701 Data Adapter Unit, IBM 2702 Transmission Control, or IBM 2703 Transmission Control.

IBM 1030 Data Collection System

IBM 1050 Data Communication System

IBM 1060 Data Communications System

IBM 2260 Display Station - IBM 2848 Display Control (Remote-attached through the 2701 only)

IBM 2740 Communications Terminal and IBM 2760 Optical Image Unit (optional attachment)

AT&T 83B3 Selective Calling Stations

Western Union Plan 115A Outstations

BTAM supports the following devices on switched networks attached to a multiplexor channel through an IBM 2701, 2702, or 2703:

IBM 1050 Data Communication System

IBM 2740 Communications Terminal and IBM 2760 Optical Image Unit (optional attachment)

AT&T Model 33/35 Teletypewriter Exchange Terminal

BTAM supports the following Audio Response Units attached directly to a multiplexor channel:

IBM 7770 Audio Response Unit

IBM 7772 Audio Response Unit

BTAM supports the following device attached directly to either the selector or multiplexor channel:

IBM 2260 Display Station - IBM 2848 Display Control (Local)

Binary Synchronous Communication Support

For binary synchronous transmission, one of the following telecommunications control units must be attached to the System/360 channel:

IBM 2701 Data Adapter Unit with Synchronous Data Adapter-Type II (SDA-II). The 2701 may be attached to either the multiplexor (switched or nonswitched lines) or the selector channel (nonswitched lines only).

IBM 2703 Transmission Control Unit with Synchronous Base I. The 2703 may be attached to the multiplexor channel only.

Binary synchronous communication is supported for the following configurations:

1. Point-to-point, nonswitched lines:

   IBM System/360* to IBM System/360*

   IBM System/360* to IBM 2780 Data Transmission Terminal

   IBM System/360* to IBM 1130 Computing System

   IBM System/360* to IBM System/360 Model 20

2. Point-to-point, switched lines:

   IBM System/360* to IBM System/360*

   IBM System/360* to IBM 2780

   IBM System/360* to IBM 1130

   IBM System/360* to IBM System/360 Model 20

3. Multipoint, nonswitched lines:

   IBM System/360* to multidropped IBM 2780

   IBM System/360* to multidropped IBM 1130

   IBM System/360* to IBM System/360 Model 20

Note: Only identical devices may be multidropped on the same line.

Machine Requirements

BTAM operates on a System/360 having at least 32K bytes of main storage. The only

----

*The System/360 may be a Model 30, 40, 50, 65, 67(in 65 mode), or 75.

additions to the minimum requirements of the IBM System/360 Disk Operating System are:

- The IBM 1052 Printer-Keyboard is mandatory.
- All telecommunications devices except the 2260 Local and the IBM 2701 with Synchronous Data Adapter - Type II (leased lines only) must be attached to the System/360 via the multiplexor channel.
- No device may be operated in burst mode on the multiplexor channel concurrently with the operation of BTAM except when the BTAM operation involves only the 2260 Local.

Prerequisites for Using this Publication

The prerequisites for a thorough understanding of this publication are a basic knowledge of System/360 machine concepts and a basic knowledge of the operations of the devices.

The prerequisite publications are:

1. IBM System/360 Principles of Operation, Form A22-6821.
2. IBM System/360 Disk Operating System, Data Management Concepts, Form C24-3427.
3. IBM System/360 Disk Operating System, Supervisor and Input/Output Macros, Form C24-5037.
4. IBM System/360 Disk and Tape Operating Systems, Assembler Specifications, Form C24-3414.
5. IBM System/360 Disk Operating System, System Control and System Service Programs, Form C24-5036.
6. IBM System/360 Disk and Tape Operating Systems, Concepts and Facilities, Form C24-5030.
7. IBM System/360 Disk Operating System, System Generation and Maintenance, Form C24-5033.
8. IBM System/360 Disk Operating System, Operating Guide, Form C24-5022.
9. IBM System/360 Disk Operating System, Performance Estimates, Form C24-5032.

Some device-oriented publications are:

1. Data Communications Concepts and Communications Facilities, Form E20-8158.
2. IBM 2701 Data Adapter Unit, Principles of Operation, Form A22-6864.
3. IBM 2702 Transmission Control, Form A22-6846.
4. IBM 2703 Transmission Control, Form A27-2703.
5. IBM System/360 Component Description, IBM 7770 Audio Response Unit, Models 1, 2, and 3, Form A27-2712.
6. IBM System/360 Component Description, IBM 7772 Audio Response Unit, Form A27-2711.
7. IBM 7772 Audio Response Unit Vocabulary, Form A27-2710.
8. IBM 1030 Data Collection System, Form A24-3018.
9. IBM System/360 Disk Operating System, Vocabulary File Utility Program for the IBM 7772 Audio Response Unit, Form C27-6924.
10. IBM 1050 Operator's Guide, Form A24-3125
11. IBM 1050 System Operations Reference Manual, Form A24-3020.
12. IBM 1050 Data Communication System Original Equipment Manufacturer's Information, Form A24-3143.
13. IBM 1060 Data Communications System, Form A24-3034.
14. IBM System/360 Component Description, IBM 2260 Display Station, IBM 2848 Display Control, Form A27-2700.
15. IBM 2740 Communications Terminal, Form A24-3403.
16. IBM 2780 Data Transmission Terminal Component Description, Form A27-3005.
17. IBM 1130 Functional Characteristics, Form A26-5881.
18. IBM 1130 Synchronous Communications Adapter Subroutines, Form C26-3706.

The following publication is a prerequisite introduction to the data link (or line) control procedures for persons who intend to use the DOS/BTAM support for Binary Synchronous Communication.

19. General Information - Binary Synchronous Communications, Form A27-3004.

Note: This publication describes the current release of DOS/BTAM. In addition, it includes a description of the following for planning purposes only: System/360-to-2780 USASCII binary synchronous support, System/360-to-System/360 Model 20 support, information on the READ Stop operation, and support for the IBM 2760 Optical Image Unit that can be attached to the IBM 2740. Refer to IBM SRL Bibliography - Supplement Teleprocessing, Form A24-3089, for other related publications, including new material on binary synchronous communication.

The Basic Telecommunications Access Method (BTAM) controls transmission and reception of messages over telecommunications lines in response to READ, WRITE, and CONTROL macro instructions issued in the user's problem program. The primary functions of BTAM are channel program generation at object time and, at the user's option, buffer management.

The facilities of BTAM are made available through the macro-generation capabilities of the System/360 Disk Operating System (DOS) assembler. The executable BTAM routines and certain information about the terminals and lines to be used are included in the Disk Operating System at system generation time. During assembly of a problem program, macro instructions coded by the user are expanded into:

1. BTAM Logic Module.
2. In-line instructions, including linkages to the executable BTAM routines.
3. Tabular information defining the lines, terminals, and options to be used.

At execution time, the problem program communicates with BTAM through various BTAM macro instructions in order to request such Tele-processing services as preparing communication lines for use, performing data transmission, or providing buffer management facilities. Preparing the lines for subsequent data transmission is accomplished by the execution of an OPEN macro instruction. OPEN will generate and execute channel programs, where required, in order to initialize the associated control unit and data set. Optionally, OPEN will also organize and format a buffer pool.

After OPEN, a message may be sent or received by executing a READ or WRITE macro instruction, which causes a branch and link to the BTAM READ/WRITE/CONTROL routine. This routine builds a channel program to perform the requested operation and passes a request for execution to the DOS supervisor, which schedules the channel program execution. Control passes back to the user at this point, and execution of the channel program is performed asynchronously with respect to problem program execution.

An important feature of BTAM is repeated restarting of channel programs in response to conditions of the line. This allows a single READ macro instruction to successively poll (or invite to send) a number of terminals on a line and to continue this polling process (without interfering with the problem program) until a message has been read. A single WRITE can signal a number of terminals to prepare to receive and then send them the message.

BTAM buffer management maintains buffer pools, providing buffers from the pool to the user as he requests them, and returning buffers to the pool when he releases them. Buffer management is performed synchronously with respect to the execution of the problem program.

CONCEPTS AND TERMINOLOGY

This section describes line and terminal configurations and message control considerations. It defines some terms used in this publication.

In general, a telecommunications system has consisted of a number of input, output, or input/output devices, usually in geographically dispersed locations, connected by one or more communication lines. A telecommunications system operating under an IBM access method, such as BTAM, and the Disk Operating System (DOS), is more specifically defined as a network of terminals connected to a central computer by one or more communication lines. In addition BTAM, used with DOS, supports CPU-to-CPU binary synchronous communication in which neither end of the communication line (switched or nonswitched) can be reasonably described as a terminal as that term has been defined in the telecommunications literature. In this publication the meaning of telecommunications system is broadened to include both remote and local terminals connected to a central computer and CPU-to-CPU communication.

A terminal is the unit or units of equipment that accepts keyed or punched data as input for sending to the computer and/or produces printed, punched, or visually displayed data or Audio messages as output received from the computer. All messages from one terminal to another pass through the computer. In addition, the computer itself may receive from the terminals and originate messages for the terminals.

A terminal consists of a control unit and one or more input/output devices. Each such device is called a component. Each input device and each output device is considered a separate component, regardless of whether they are physically combined. For example, an IBM 1050 is referred to as a terminal; its constituent devices or components include the 1053 Printer, 1054 Paper-Tape Reader, keyboard section of the 1052 Printer-Keyboard, printer section of the 1052 Printer-Keyboard, etc.

Terminals in a telecommunications system operated under BTAM control are usually separated from the computer by a distance sufficient to require common-carrier facilities and transmission techniques to accomplish communication with the computer. The system may, however, include local terminals that are located on the same premises as the computer and attached directly to the channel. The important distinction, however, is the way in which the cable or lines are attached to the computer. Terminals that are connected through a 2701, 2702, or 2703 telecommunications control unit are all classified as remote. Units that are directly connected to a System/360 channel are classified as local.

In communications terminology, the terms communication line, data link, data path, circuit, and channel are somewhat synonymous. All of these terms are used to represent the medium by which the physical components of a system are connected. In this publication the term communication line (or line) is used, and, except as otherwise qualified, refers to any medium, whether it be a telegraph circuit, a telephone circuit, a privately owned line, etc.

Each remote terminal is connected to a communication line by either a data set, or a modem (modulator/demodulator), etc., depending on the kind of communication line and the kind of terminal involved. The precise functions vary, but the overall purpose is the same: to provide an interface between terminal and line. This publication uses the term data set to represent any of these units (not to be confused with a program data set, which is an Operating System term).

The programmer generally need not concern himself in any way with these data sets, because their presence generally exerts no influence on programming. They are defined in the interest of providing a complete, accurate picture of the line and terminal configuration.

The term station means the aggregate of equipment and controls attached to any one of the several possible ends of a communication line. Put another way, a station is a terminal (including the terminal components) plus the equipment by which the terminal is attached to the line. Even more generally, the meaning of station may be broadened to include either CPU in a CPU-to-CPU communication system.

Telecommunications Networks

A telecommunications system may include a switched network or a nonswitched network, or a combination of the two.

A nonswitched network consists of one or more private or leased lines that connect the computer to one or more remote units. The computer and the remote units are physically connected; that is, the circuits making up the communication lines are continuously established for predetermined time periods during which data transmission may proceed between the computer and the remote stations. In this type of system, the computer can, under certain conditions, send messages to more than one remote device on the same line at the same time. These lines comprising a nonswitched network are known as private or leased lines. Leased lines are furnished by a common carrier on a contract basis, between specified locations for a continuous period or regularly recurring periods at stated hours, for the exclusive use of one customer.

A switched network consists of a number of remote stations with which the computer can communicate. The computer and the several remote device are each continuously connected by access lines to the common-carrier exchanges serving their respective locations. A continuous data path is established between computer and remote device only for the period of time transmission is to take place. The connection is established by dialing the telephone number of the unit (either terminal or CPU) at the other end. In this type of system, communication is established between the computer and only one unit at a time on each line. In this case, line refers to a discrete data path between the transmission control unit and the common-carrier exchange.

In this publication, the term switched network refers to any network in which the data link between computer and remote device must be established by dialing, for data transmission to occur. The term nonswitched network refers to a network in which the communication lines linking computer and remote units are continuously established, thus requiring no dialing.

Figure 1 shows a Tele-processing system with both a switched and a nonswitched network.

Message Control

A programmer using BTAM must be aware of the line-control conventions for the particular type (or types) of terminal and line configurations for which he is programming. This section deals with general considerations about line control. Specific information about each type of terminal and line configuration is given in later sections of this publication.

In some respects, the functions that must be performed by the programmer using BTAM are parallel with those performed in telecommunications systems that are not computer-oriented. In any telecommunications system, contact between terminals must be established before a message is sent. In some systems, terminals wishing to send a message contend with one another for use of the line. The first terminal to initiate contact on a line that is not currently in use controls the line and prevents its use by other terminals until it has concluded its message transmission. A system operated in this manner is a contention system.

In other noncomputer oriented systems, one of the terminals is specified as the control station. This initiates all contacts for all other terminals on the line, using a procedure known as polling. Polling is a flexible, systematic, centrally controlled method permitting terminals on a multiterminal line to transmit without contending for use of the line. The control station contacts the other terminals periodically and invites them to send any messages they have ready. In addition, the control station itself may elect to send. A system operated in this manner is a polling system.

Polling is accomplished by sending on the line one or more selection characters. In some cases, two characters are used: the first selects the terminal, the second selects the specific component of that terminal.

In centralized operation, the terminal identified by these characters then sends a response to the control station -- a message if it has a message to send, a negative response if it does not. The control station may poll a number of terminals and components, in turn, until one is found that has a message ready. Similarly, when the control station terminal has a message to send, it transmits on the line one or more addressing or call-directing characters. As in polling, two characters are often used; the first selects the terminal, the second selects the component. The terminal identified by these characters returns a response. It returns a positive response if it is able to accept the message; a negative response, if it is not.

In noncentralized operation, a terminal that is polled by the control station may send a message directly to another terminal without involving the control station in a message-switching operation.

A telecommunications system operated under BTAM is usually a polling system in which the role of control station is played by the computer. Moreover, it is a centralized system, that is, terminals send their messages not to other terminals, but to the computer. The computer then relays the messages to the appropriate destination terminals. Polling is not used, however, for the 2260 Local, the audio response units or the 2740 without station control.

The polling and addressing functions are performed in both switched and nonswitched systems, with minor variations.

In a switched network, the line connection must be completed between computer and remote device before message transmission can proceed. The connection may be established by either the computer or remote unit. When the computer wishes to establish the connection, it dials the telephone number of the remote unit. The connection becomes established when the unit responds. The functions performed by the computer in this case are known as calling. Polling, addressing, or ID Verification, if needed, may then take place.

Ordinarily, the computer calls a terminal only when it wishes to address the terminal (to send it a message), rather than to poll it (to solicit messages). When a terminal wishes to establish the connection, the person operating the terminal dials the computer's telephone number (or one of its several numbers). The connection is established when the computer responds. The function performed by the computer in this case is known as answering. Polling or addressing may then take place.

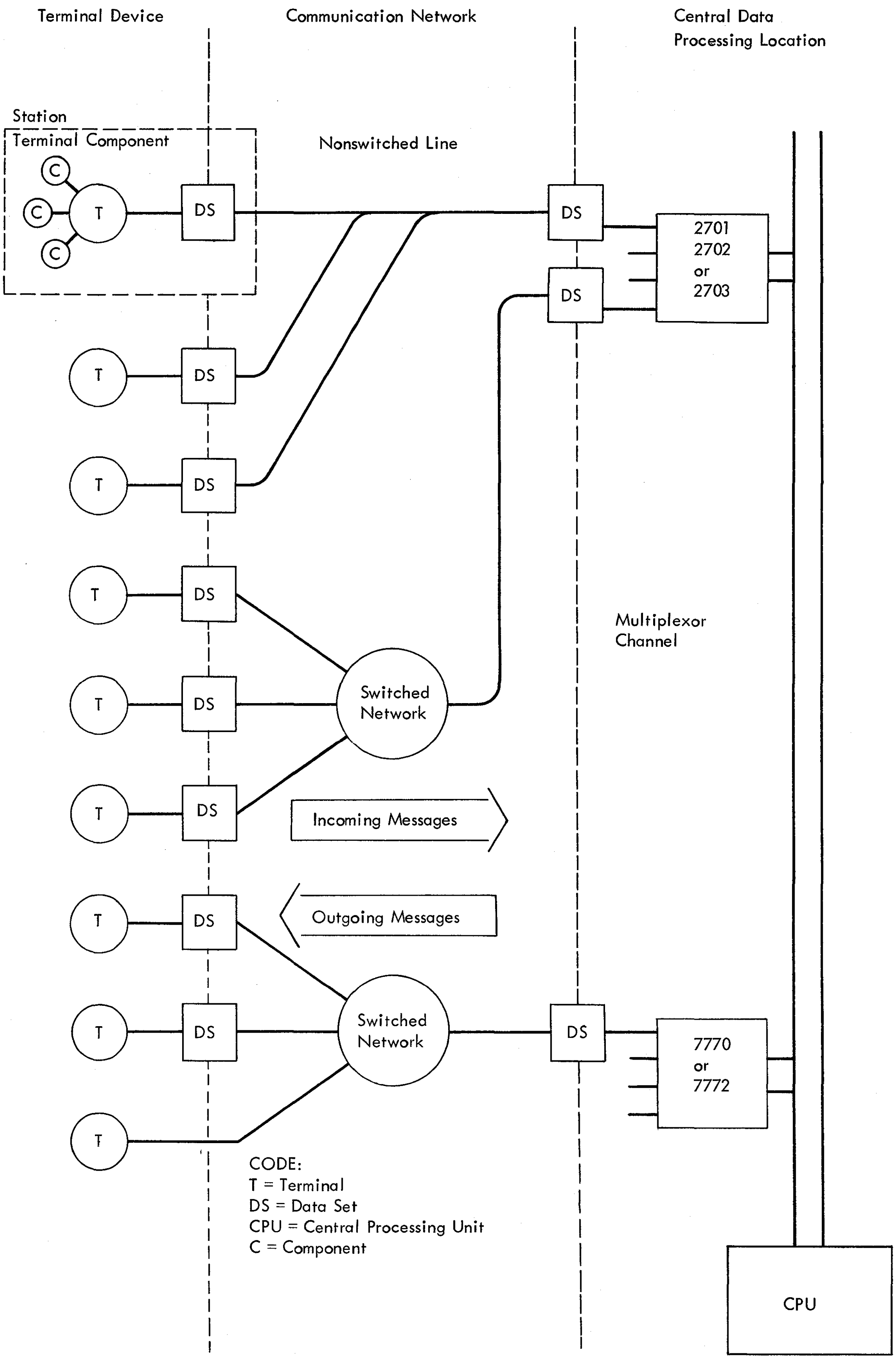


Figure 1. Configuration of a Communication System (Part 1 of 2)
Start-Stop System

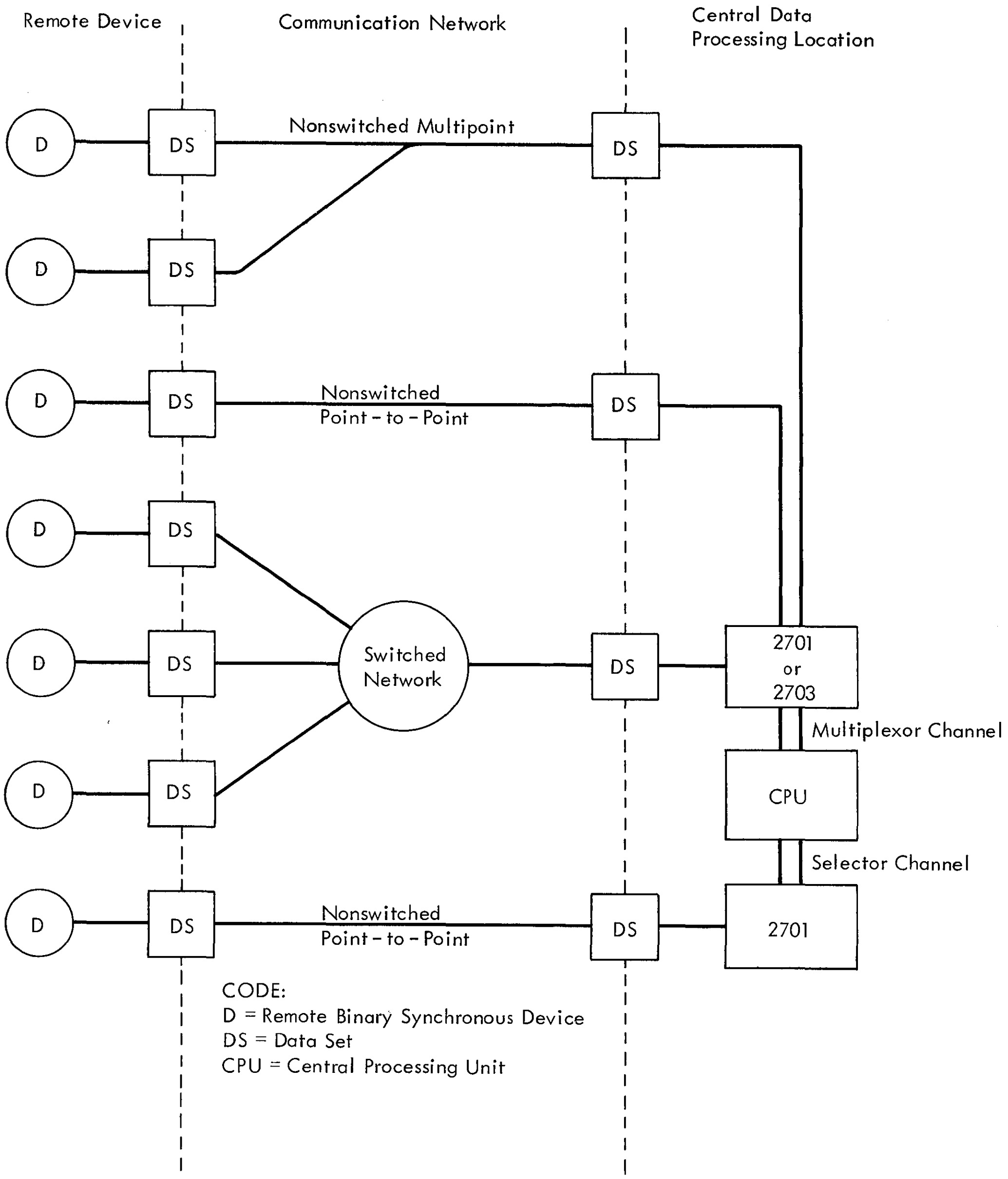


Figure 1. Configuration of a Communication System (Part 2 of 2) Binary Synchronous System

Ordinarily, a terminal calls the computer only when it wishes to be polled for a message it has ready for the computer or another terminal. Note that regardless of which party -- computer or terminal -- establishes the line connection, message flow from the terminal to the computer is achieved by reading from the terminal, and message flow from the computer to a terminal is achieved by writing to the terminal except for those devices where polling/addressing is not used.

Although terminals can call the computer at any time, the computer, to fulfill its function as control station, must be able to accept or reject incoming calls. Therefore, the computer performs a function known as <u>enabling of the line</u>. <u>Enabling</u> is the process of conditioning the telecommunications control unit (TCU) and the data set to accept incoming calls on a line. The user's program determines which lines are, at a given moment, to be enabled, and which are not. If a terminal calls in on a line that is currently enabled, and that is not in contact with another terminal, the line connection is completed and message transmission (which may be preceded by polling or addressing) can occur. If a terminal calls in on a line that is not currently enabled, or that is enabled but is occupied with another terminal, the calling terminal receives a busy signal, and contact is not established. The terminal must wait and call again later.

Calling and answering functions are needed in CPU-to-CPU communication when line connection is over a switched network. Although polling and addressing are not used, the programmer may elect to employ BTAM-supported ID verification procedures which provide greater security protection and line assurance. The section, CPU-To-CPU Dial, discusses these procedures in detail.

In a nonswitched network, the line connections between computer and terminals are continuously established; hence, the calling and answering are not required. Only the computer can initiate contact with remote terminals except in contention operation involving the 2740 or the 2260 Local. As in switched networks, the computer polls terminals to solicit messages from them and addresses terminals to send messages to them.

Such functions as polling, addressing, calling, answering, and ID Verification, while performed by BTAM, require that the programmer supply the appropriate characters to be used and issue the BTAM macro instructions performing these functions. There are, however, other functions automatically provided by BTAM and about which the programmer generally need not concern himself. For example, when the problem program sends a message via BTAM to a remote device, the remote device generally responds with a control character(s), signifying negative or positive acknowledgement of the message (the negative or positive acknowledgement indicating the message was received with or without I/O error). Such control characters are analyzed by BTAM and further action may be initiated by BTAM. In the case in which a negative acknowledgment is received, for example, BTAM will automatically initiate a retransmission when the Error Recovery Procedures are included in the BTAM module.

There are other functions with which the programmer does need to concern himself. For example, the Start-of-Text (STX) control character is generally required as the first character to be sent to the remote device. In later sections of this publication various device-dependent control character sequences are discussed in greater detail (see Device Dependent Considerations).

In binary synchronous communication these line control procedures are standardized and are generally independent of the remote device being used, insofar as the basic functions are concerned. For example, alternating acknowledgments are implemented in response to messages sent and/or received, regardless of the remote device used. Alternating acknowledgments ACK-0 and ACK-1 (even and odd) are used, as the name suggests, to acknowledge (positively) alternate messages. For instance, if a response from the remote device to a message sent by the CPU is ACK-0 (even), the (positive) response to the next message sent by the CPU will be ACK-1 (odd). This affords the programmer greater message assurance: protection against duplication of or loss of message blocks.

It is necessary for the programmer to understand these functions, insofar as message processing is required to recover from certain error conditions as posted by BTAM. For example, if the wrong positive acknowledgement is received (say, ACK-0 is received in a response to an odd message block), BTAM automatically attempts recovery. After a fixed number (as specified in the DTFBT macro by the user) of unsuccessful retries, BTAM posts the error condition and control is returned to the user's program with further action determined by the user.

In the more typical case, a time out may occur (no acknowledgment received) after a message has been transmitted. In this case, BTAM automatically sends the inquiry signal (the ENQ control character) requesting the remote device to retransmit its last-sent acknowledgment. If this is the wrong acknowledgment, BTAM automatically retransmits the message and recovers from the error condition without intervention by the user's program.

The use of alternating acknowledgments and other data link control characters is shown in the Device Dependent Considerations section of this publication.

Sharing the Processing Time

The processing time in a multiprogramming environment may be shared by two types of problem programs: background and foreground. Background programs are initiated by job control from the batched-job input stream. Foreground programs are initiated by the operator from the console-typewriter. Foreground programs do not execute from a stack. When one completes, the operator must explicitly initiate the next foreground program.

Background and foreground programs are initiated and terminated asynchronously with each other. Neither is aware of the status or existence of the other.

A telecommunications program may be run as either a foreground or a background program. Normally, it is run as a foreground-one program to have the highest priority of any program being executed at a particular time. Normally, a background program is run along with the telecommunications program. If 10K of main storage is not available for job control to initiate a background program, the telecommunications program must be run as a background program.

DOS is capable of concurrently operating one background program and one or two foreground programs. Priority for CPU processing is controlled by the supervisor, with foreground programs having priority over background programs. When an interruption occurs, the supervisor gains control, processes the interruption, and gives control to the highest priority program that is in a ready state. Control is taken away from the highest priority program when that program must wait for the completion of a specific event before further processing can be performed. Control is taken away from a lower priority program when an event on which a higher priority program has been waiting has been completed, or when that program must wait for the completion of a specific event before further processing can be performed. When all programs in the system are simultaneously waiting (i.e., no program can process), the system is placed in the wait state, enabled for interruptions. When an interruption satisfies a program's wait condition, that program becomes active and competes with other programs for CPU processing time.

Figure 2 shows the general functions provided for a problem program operating under DOS/BTAM. Issuance of a READ/WRITE/CONTROL starts at numeral one (1); interruption processing starts at numeral nine (9).

Communication Between the Problem Program, BTAM, and the Supervisor

1. The problem program communicates requests for I/O operations and other services to BTAM through the BTAM macro instructions. The information needed by BTAM to perform these functions is provided directly by the macro operands and by tables pointed to via these operands. For example, for I/O operations, the DTFBT (a table containing information defining the line group) and the DECB (a block containing information needed to generate a channel program) are required by BTAM.

2. BTAM requests input/output operations from the supervisor, as do other DOS access methods. See IBM System/360 Disk Operating System, Supervisor and Input/Output Macros, Form C24-3429. An Execute Channel Program macro instruction (EXCP) specifies a Command Control Block (CCB) as an operand. The CCB contains the address of the channel program to be executed. Under BTAM the CCB is assembled as a part of a Line Control Block (LCB). The LCB contains other fields in which BTAM maintains control information for the line as well as an area in which the channel program is generated. Each line in a line group is associated with a separate LCB. The DTFBT, in turn, incorporates all the LCB's for the line group as well as other fields containing control information for the line group as a whole. Because the LCB is used by BTAM logic, it must not be used or modified by the problem program.

3. I/O interruptions are asynchronous with the problem program. BTAM posts each line activity completion in the Data Event Control Block (DECB) associated with that line. When further processing is contingent on the termination of I/O operations, the user releases control by a WAIT macro instruction. When that WAIT is satisfied, control returns to the problem program at the instruction following the WAIT.

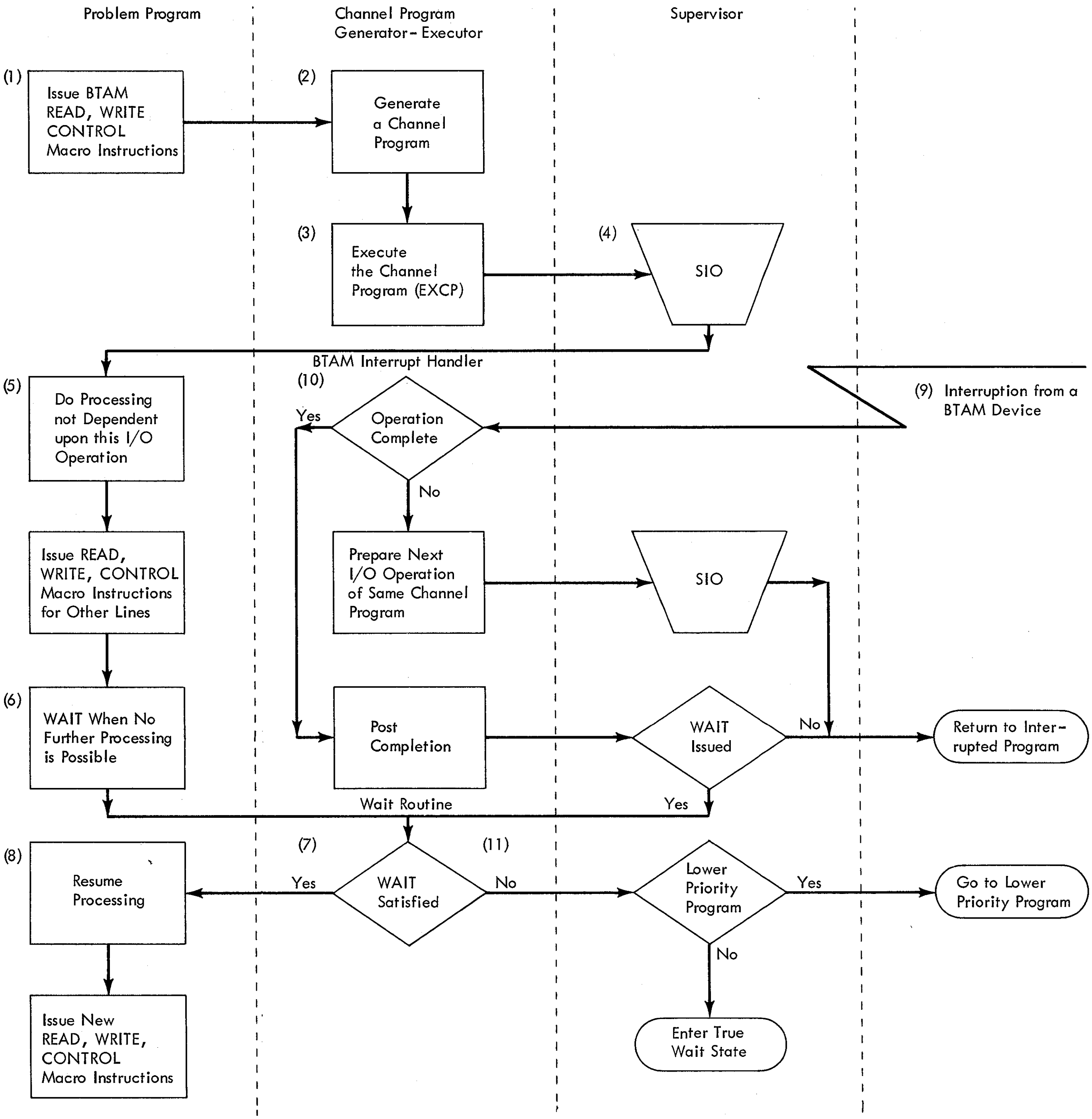

Problem Program
Channel Program Generator- Executor
Supervisor
(1)
Issue BTAM READ, WRITE CONTROL Macro Instructions
(2)
Generate a Channel Program
(3)
Execute the Channel Program (EXCP)
(4)
SIO
BTAM Interrupt Handler
(10)
Operation Complete
Yes
No
(9) Interruption from a BTAM Device
(5)
Do Processing not Dependent upon this I/O Operation
Issue READ, WRITE, CONTROL Macro Instructions for Other Lines
Prepare Next I/O Operation of Same Channel Program
SIO
(6)
WAIT When No Further Processing is Possible
Post Completion
WAIT Issued
No
Return to Interrupted Program
Yes
Wait Routine
(7)
(11)
(8)
Resume Processing
Yes
WAIT Satisfied
No
Lower Priority Program
Yes
Go to Lower Priority Program
No
Issue New READ, WRITE, CONTROL Macro Instructions
Enter True Wait State

Figure 2. DOS/BTAM General Functions

The user assembles the appropriate BTAM routines according to the telecommunications requirements of his problem program. The BTAM module assembled includes routines for:

1. READ, WRITE, and CONTROL
2. Multiple wait and RESETPL
3. I/O interruption processing
4. REQBUF and RELBUF (optional)
5. Buffer Logic for READ/WRITE (optional)
6. Error recovery procedures (optional except for binary synchronous communication and Audio Response units)
7. Error Counts (optional)
8. Code Translation (optional)
9. On-line terminal testing (optional)
10. The 2260 Local support (optional)
11. The 7770/7772 support (optional)
12. Switched network support (optional)
13. Binary Synchronous Communication logic (optional)
14. Start-stop Auto Poll support (optional)
15. BSC multipoint logic (optional)

The macro instructions provide the user with the means of control over BTAM and can be broken down into four categories:

1. Declarative macro instructions are used to generate lists containing terminal and line information. They are also used to create or load tables, and to define the BTAM module. They are assembly-time macro instructions.

   BTMOD The BTMOD macro instruction permits the user to select the options to be included in the BTAM module.

   DTFBT The DTFBT macro is the user's means of defining a line group and selecting any optional features of BTAM that are to be utilized for that line group.

   DTFBTND The DTFBTND macro instruction has no function in the present version of BTAM; it is included here to maintain compatability with the previous version of BTAM.

   DFTRMLST The DFTRMLST macro instruction creates terminal lists that contain the characters necessary to perform polling, addressing, dialing, automatic answering or ID checking.

   LERB The LERB macro generates and initializes a table for the accumulation of error counts.

   ASMTRTAB The ASMTRTAB macro instruction loads the translation table(s) requested by the user.

2. Transient macro instructions activate or deactivate one or more line groups and are executed in the transient area.

   LOPEN The LOPEN macro instruction is used to condition a line on which an error occurred during OPEN, and when necessary, initiates line delays.

   OPEN The OPEN macro instruction prepares communication line groups for use, and, when necessary, OPEN generates and executes required channel programs. Optionally, it may also organize a buffer pool.

   CLOSE The CLOSE macro instruction removes communication line groups from use. Active channel programs are halted, if necessary.

3. Data-handling macro instructions control the communication lines, perform data transmission, terminate polling, change terminal lists, perform code translation, etc., at execution time.

   READ The READ macro instruction updates the DECB, generates linkage to the BTAM READ/WRITE/CONTROL routine, and causes BTAM to perform a specified function (such as polling terminals or reading a message into an input buffer area). READ may also be used to generate a DECB at assembly time.

WRITE The WRITE macro instruction updates the DECB and causes BTAM to perform a function, such as addressing a terminal and writing a message to it from an output area. WRITE may also be used to generate a DECB at assembly time.

CONTROL The CONTROL macro instruction causes BTAM to execute a control function, i.e., enabling or disabling a line. May also be used to generate a DECB at assembly time.

CHGNTRY The problem programmer may use the CHGNTRY macro instruction to deactivate or activate a polling or addressing entry in a terminal list without redefining the list.

RESETPL The RESETPL macro instruction causes a read operation to terminate by interrupting polling. On a switched-connection line, it causes the halting of an outstanding Enable command. It is also used to terminate a Prepare command.

WAIT The WAIT macro instruction causes release of control of the CPU until a user-specified number of events have been completed.

TWAIT The TWAIT macro instruction relinquishes control of the CPU until one of a number of events has been completed. Optionally the terminal test facility may be implemented by this macro instruction.

LERPRT The LERPRT macro instruction provides the ability to print the current values of the number of transmissions and error count accumulators.

TRNSLATE The TRNSLATE macro instruction causes the translation of a message block(s) from one code to another.

ONLTST The ONLTST macro is used to generate linkage to the binary synchronous on-line test logic in BTMOD.

4. BTAM buffer management macro instructions cause BTAM to perform services that are not directly connected with communication line activity.

The following table shows which macro instructions do not apply to the various BTAM-supported devices. For this reason the user must not use these macro instructions for the indicated devices.

Device	Non-Applicable Macro Instructions
1030 1050 (Nonswitched) 1050 (Switched) 1060 2260 (Remote) 83B3 115A	CONTROL
TWX 33/35	CONTROL, CHGNTRY
7770/7772	DFTRMLST, CHGNTRY, LERB, RESETPL, LERPRT, LOPEN
2260 (Local)	CONTROL, DFTRMLST, LERB, LERPRT
2740 (Basic)	CONTROL, DFTRMLST, CHGNTRY, RESETPL
2740 D (Dial)	CONTROL, CHGNTRY, RESETPL
2740 S (Station Control)	CONTROL
2740 SC (Station Control and Checking)	CONTROL
2740 DTC (Dial, Transmit Control and Checking)	CONTROL, RESETPL
2740 C (Checking)	CONTROL, DFTRMLST, CHGNTRY, RESETPL
2740 DC (Dial and Checking)	CONTROL, CHGNTRY, RESETPL
2740 DT (Dial and Transmit Control)	CONTROL, RESETPL

REQBUF The REQBUF macro instruction provides one or more buffers from a buffer pool if they are available.

RELBUF The RELBUF macro instruction returns one or more buffers to a buffer pool.

This section provides a coding format illustration for each macro instruction described. Because the form of tabular data generated and/or the exact function performed varies depending on the particular remote device or network configuration, no attempt is made in this section to discuss the uses of macro instructions in detail. Rather, the section, Device Dependent Considerations, presents greater detail on the use of the macro instructions for each specific device or network configuration.

To describe macro instruction operands, a system of conventions is defined for use in the format descriptions:

- Both positional and keyword operands are described by a 3-part structure. Positional operands are described by a lowercase name followed by a hyphen followed by a value mnemonic or a coded value.

  Example: numchars-absexp

  The lowercase name, numchars, is merely a convenient referent to the operand and, along with the hyphen and value mnemonic, is never coded by the programmer. The programmer replaces the positional operand in his coding by an expression defined by the value mnemonic as allowable.

  In the case of keyword operands the 3-part structure consists of the keyword followed by an equal sign (both of which must be coded as shown) followed by a value mnemonic or coded value which describes what to code on the right side of the equal sign.

  Example: BUFCB=symbol

- Coded values are written in the format description as numbers or uppercase letters and must be coded as shown.

- Value mnemonics are written lowercase in the format descriptions and indicate how an operand is to be coded by the programmer. The value mnemonics used in this publication are:

  symbol — Any symbol that the assembler accepts in the name field of an instruction.

  relexp — A relocatable expression (acceptable as an A-type address constant by the assembler).

  code — One of the coded values defined as allowable by the individual macro.

  absexp — Any absolute expression as defined by the assembler: self-defining terms (decimal, hexadecimal, binary, character), length attributes, absolute symbols, paired relocatable terms in the same CSECT, and arithmetic combinations of absolute terms.

  integer — A decimal self-defining term.

  decchars — Concatenated decimal digits (differs from "integer" in that each digit is assembled in binary format individually).

  hexchars — Concatenated hexadecimal digits (the framing characters: X' ' are not coded by the programmer).

  (r) — Register notation specifying, by an absexp enclosed in parentheses, any register 2-12 and allowing the programmer to load the specified register with the appropriate value at execution time. Certain macro instructions also permit use of registers 0 and 1 by specifying (0) or (1). The 0 or 1 within parentheses must be coded literally.

  Note: When the format description indicates that register notation is an alternative to absexp, the programmer should be aware that an expression framed by parentheses is assumed to be register notation.

- {}Braces are used in two ways:

1. To define grouping of the alternate forms of a mandatory operand. For example:

$$\text{listype-}\begin{Bmatrix}\text{OPENLST}\\ \text{WRAPLST}\\ \text{DIALST}\end{Bmatrix}$$

2. To define a syntactical unit when the braces are followed by an ellipsis (three periods). For example:

   {dtfbt-symbol,}...

   The entire expression indicates that the syntactical unit (enclosed by braces) can be coded one or more times.

- [] Brackets indicate an optional operand. For example:

  [MSGL=absexp,]

  When one of the alternate forms of the operand is underlined, the function of that form is used if the operand is not coded. For example:

$$\begin{bmatrix}\underline{\text{SWITCH=NO,}}\\ \text{SWITCH=YES,}\end{bmatrix}$$

  If neither form is coded, SWITCH=NO is assumed.

The listed descriptive symbols are not to be coded by the programmer. On the other hand, parentheses and the apostrophes (single quote marks) which appear in the macro format illustrations must be coded as shown.

The rules for coding commas to separate operands are:

- When the programmer omits a positional operand in the coding, the trailing comma must still be coded unless:

  1. it is the last positional operand in the macro.

  2. the programmer also omits all positional operands which normally are coded following the omitted positional operand.

  3. the syntax in the macro format illustration indicates the comma may be omitted along with the operand. (This is true for the device-dependent usage of DFTRMLST. See especially the use of DFTRMLST for IDLST in the device-dependent sections of this publication.)

- When a keyword operand is omitted, its trailing comma is also omitted.

- The comma following the last coded operand is to be omitted.

Note: Comma usage is in accordance with the conventions defined for the DOS macro generator.

In addition to this section presenting the macro format descriptions, an additional summary of allowable values for the BTAM macros is provided in Appendix C.

DECLARATIVE MACROS

The declarative macro instructions are the following:

BTMOD
DTFBT
DTFBTND
DFTRMLST
LERB
ASMTRTAB

BTMOD (BTAM Module) Macro Instruction

The BTMOD macro instruction (Figure 3) provides the user with the facility to assemble the routines of BTAM to perform the telecommunications functions of his problem program. The BTAM module includes routines for:

1. READ, WRITE, and CONTROL

2. Multiple wait and RESFTPL

3. I/O interruption processing

4. REQBUF and RELBUF (optional)

5. Buffer Logic for READ/WRITE (optional)

6. Error recovery procedures (optional, except for binary synchronous communication and Audio Response units)

7. Error Counts (optional)

8. Code Translation (optional)

9. On-line terminal test facilities (optional)

10. The 2260 Local support (optional)

11. The 7770/7772 support (optional)

12. Switched network support (optional)

13. Binary Synchronous Communication logic (optional)

14. Start-stop Auto Poll support (optional)

Name	Operation	Operands

[symbol]	BTMOD	SEPASMB=NO,
		SEPASMB=YES,
		BUFFER=NO,
		BUFFER=REQREL,
		BUFFER=YES,
		ERLOGIC=E,
		ERLOGIC=N,
		ERLOGIC=C,
		ERLOGIC=NC,
		TERMTST=NO,
		TERMTST=YES,
		SWITCH=NO,
		SWITCH=YES,
		L2260=NO,
		L2260=YES,
		AUDIO=NO,
		AUDIO=YES,
		CANCEL=NO,
		CANCEL=YES,
		TRANSL=NO,
		TRANSL=YES,
		BSCS=NO,
		BSCS=YES,
		SSAPL=NO,
		SSAPL=YES,
		BSCMPT=NO,
		BSCMPT=YES,
		BSCTEST=NO
		BSCTEST=YES

•Figure 3. Format of BTMOD Macro Instruction

15. BSC multipoint logic (optional)

Note: BTMOD should not be assembled in the middle of the user's code. See Principles of BTAM Register Usage for a detailed discussion of this point.

symbol
Used to assign a name to the assembled BTAM logic module. If no symbol is coded in the BTMOD name field, a standard name (IJLBTM) is generated.

SEPASMB
Specifies how the BTAM logic module is to be assembled.

NO specifies the BTAM module is to be assembled with the problem program (see Assembly Considerations).

YES specifies the BTAM module is to be assembled separately. This causes the appropriate CATALR control card to be punched, thus permitting the user to catalog the module in the relocatable library for later retrieval by the linkage editor (see Assembly Considerations).

BUFFER
Specifies whether the BTAM module is to contain the buffer management routines, where:

NO specifies the buffer management routines are not to be included.

REQREL specifies that only the REQBUF/RELBUF logic is to be included.

YES specifies that the REQBUF/RELBUF logic and the logic to read into buffers and to write from buffers are to be included.

ERLOGIC
Specifies what error handling routines are to be included in the BTAM module, where:

E specifies (start-stop and 2260 Local) error recovery procedures (ERP) with no error count.

N specifies (start-stop and 2260 Local) error recovery procedures are not to be included.

C specifies both (start-stop and 2260 Local) ERP and error count are to be included.

NC specifies error count logic is to be included but (start-stop and 2260 Local) ERP is to be excluded.

Note: This operand does not apply to Audio Response Units. The ERP unique to the 2260 Local is always omitted if L2260=NO is coded.

TERMTST
Specifies whether the on-line terminal test facility is to be included, where:

NO specifies this facility is not to be included in the BTAM module.

YES specifies this facility is to be included in the BTAM module (see TWAIT macro instruction).

SWITCH
Specifies whether the BTAM logic concerning switched networks is to be included, where:

NO specifies this logic is not to be included in the BTAM module.

YES specifies this logic is to be included in the BTAM module.

L2260
Specifies whether the BTAM logic module for the 2260 Local is to be included, where:

NO specifies this logic is not to be included in the BTAM module (2260 Local support not needed).

YES specifies that the logic is to be included.

AUDIO
Specifies whether the BTAM logic for support of the audio response units is to be included, where:

NO specifies this logic is not to be included in the BTAM module (7770/7772 support not needed).

YES specifies this logic is to be included in the BTAM module.

Note: If AUDIO=YES is coded, the error recovery procedures unique to audio are automatically included.

CANCEL
Specifies whether the BTAM CANCEL logic will cancel a job or return control to the user when a cancel condition is detected.

NO specifies that control should be returned to the user at the instruction following the BTAM macro instruction which detected the cancel condition, along with the conditions for cancellation.

YES specifies that a DOS CANCEL macro instruction is given when a cancel condition occurs.

TRANSL
Specifies whether the Code Translation facility is to be included, where:

NO specifies this facility (see TRNSLATE macro instruction) is not to be included.

YES specifies this facility is to be included.

BSCS
Specifies whether the BTAM logic for point-to-point binary synchronous communication is to be included where:

NO specifies this logic is not to be included.

YES specifies this logic is to be included. In addition, the ERP unique to binary synchronous communication is automatically included, independent of the ERLOGIC operand. Only ERLOGIC=C and ERLOGIC=NC are applicable to binary synchronous communication, since the error count logic is the same for start-stop and synchronous lines.

SSAPL
Specifies whether the BTAM logic for the start-stop Auto Poll support is to be included, where:

NO specifies that this logic is not to be included in the BTAM module.

YES specifies that this logic is to be included in the BTAM module.

BSCMPT
Specifies whether the logic required to support multidropped BSC devices is to be included in the BTAM module, where:

NO specifies this logic is to be excluded from the assembled BTAM module.

YES specifies this logic is to be included in the assembled BTAM module. This option requires that BSCS=YES also be coded in BTMOD.

BSCTEST
Specifies whether the logic for the BSC on-line terminal test is to be included in the BTAM module, where:

NO specifies this logic is not to be included in the assembled BTAM module.

YES specifies that this logic is to be included in the assembled BTAM module. This option requires that BSCS=YES also be coded in BTMOD.

DTFBT (Define the Line Group for BTAM) Macro Instruction

The DTFBT macro instruction (Figure 4) permits the user to define a line group and, if necessary, associated buffers through tne keyword operands of tne macro. In the case of the 2260 local, a DTFBT must be defined for each 2848.

A line group is a group of communication lines that meet the following operational requirements:

1. All lines in the group are attached to the channel through the same type of telecommunications control unit.

   Example: 2701

2. Line connection between the control unit and the remote devices is of the same type.

   Example: switched network

3. All devices within the same line group have the same line control procedures, transmission code, and operating characteristics.

4. All lines within a group share the same buffer pool if one is defined.

LINELST
    Specifies via a sublist the correspondence between symbolic unit (SYSnnn) and relative line number. In the sublist the user codes one 3-digit number for each line in the line group. The 3-digit number is interpreted as the 'nnn' of SYSnnn. The order of coding the 3-digit numbers determines which symbolic units are associated with the individual lines in the line group. As many as thirty-one 3-digit numbers from 000-244 may be coded in the sublist.

    Example: LINELST=(005,010,007)
    This results in associating:

        SYS005 with relative line number 0,
        SYS010 with relative line number 1,
        and
        SYS007 with relative line number 2

    in a line group comprising 3 lines.

SWITCH
    Specifies the type of line connection between the system and the remote device, where:

    NO specifies the line connection is permanent.

    YES specifies the line connection is through a switched network.

CU
    Defines the control unit attached to the channel as a 2701, 2702, 2703, 7770, 7772, or 2848.

```
Name      Operation    Operands

symbol    DTFBT        LINELST=([nnn-integer,]...),  [SWITCH=NO, ]
                                                     [SWITCH=YES,]

                       CU=code, DEVICE=code,

                       [FEATURE=(device dependent sublist),]

                       [BUFCB=symbol, [BUFNO=absexp, BUFL=absexp,]]

                       [SEPASMB=NO, ]  [MODNAME=IJLBTM,]
                       [SEPASMB=YES,]  [MODNAME=symbol,]

                        [MSGL=absexp,] [TERMTST=NO, ]
                                       [TERMTST=YES,]

                       [ERROPT=E, ]
                       [ERROPT=N, ]
                       [ERROPT=R, ]    [LERBADR=symbol,] [MODELST=(sublist),]
                       [ERROPT=W, ]
                       [ERROPT=RW,]

                       [          {EBCDIC}]
                       [CTLCHAR= {ASCII }],  [RETRY=absexp,] [CONFIG=PPT]
                       [          {TRNSCD}]                  [CONFIG=MPT]
```

Figure 4. Format of DTFBT Macro Instruction

Example: CU=2701

DEVICE
Defines the device as a 1030, 1050, 1060, 2260, 83B3, 115A, TW33, TW35, 2740, 2848, S360 (System/360), 2780, 1130, or 2020.

Notes: DEVICE=2260 applies to the 2260 Local.

DEVICE=2848 applies to the 2260 Remote.

DEVICE=S360 applies to a remote System/360. The user may, however, include point-to-point lines to the 2780 or to the 1130 in the line group, provided the other operands in DTFBT are consistent in their application to these lines and provided that he does not attempt to use READ/WRITE options which are not applicable to the 2780 or to the 1130 but which are available for System/360 lines.

DEVICE=2780 applies to a remote 2780 on a point-to-point or multipoint line.

DEVICE=1130 applies to a remote 1130 on a point-to-point or multipoint line.

DEVICE=2020 applies to a remote Model 20 on a point-to-point ior multipoint line.

This operand must be omitted for the 7770 and 7772.

FEATURE
Specifies, via a sublist, various device-dependent hardware and software special features. The features are specified by coding in the sublist the following appropriate codes:

PCI May only be coded when the control unit is the 7772 (see Device Dependent Considerations).

STC Specifies that the 2740 is equipped with the station control feature.

CHK Specifies that the 2740 is equipped with the checking feature.

TRC Specifies that the 2740 is equipped with the transmit control feature.

BSC Must be coded when the DEVICE operand specifies a binary synchronous device.

KBL Specifies that the 2848 control unit is equipped with the data entry feature.

Note: The Lock option optypes are rejected as undefined when the FEATURE keyword operand is not specified as KBL.

OIU Specifies that a 2760 Optical Image Unit is attached to an IBM 2740 with the checking feature and (optionally) the dial feature.

SIX
SXW
SIW If ID Verification is to occur in the CPU-to-CPU Dial system or the CPU-to-Model 20 Dial system for a calling operation, one only of these three codes must appear in the sublist. A more detailed explanation of these codes is given in the section: Device Dependent Considerations (CPU-to-CPU Dial).

RIX
RXW
RIW If ID Verification is to occur in the CPU-to-CPU Dial system or the CPU-to-Model 20 Dial system for an answering operation, one only of these three codes must appear in the sublist. A more detailed explanation of these codes is given in the section: Device Dependent Considerations (CPU-to-CPU Dial).

MAS
SLV Either MAS (Master) or SLV (Slave) may be coded in the sublist to specify whether the CPU is to be Master or Slave when contention occurs in the CPU-to-CPU Contention system (private line). If neither is coded, MAS is assumed for this system. (See Device Dependent Considerations.) Note: If MAS is specified, the remote device is to be the slave when contention occurs. If SLV is specified, the remote device is the master. When the remote device is the 2780, SLV must always be coded. The CPU must always be the slave when contention occurs between the CPU and the remote 2780.

APL Specifies that the Auto Poll feature is to be employed for the start-stop devices 1030, 1050 (non-switched), 1060, 2740S, or 2740SC. If omitted, the generated channel programs for these devices will perform the standard programmed polling. When the Auto Poll feature is specified for the 2740, the station control feature must always be present but the checking feature is optional. In the example below, the FEATURE keyword operand indicates the coding for a 2740 when the Auto Poll feature is coded. When the Auto Poll feature is coded for any of the supported devices, only CU-2702 or 2703 may be coded. If CU=2702 is coded along with APL in the FEATURE sublist, the 2702 must be equipped with the Auto Poll feature. If the 2702 is not equipped with the Auto Poll feature, a command reject error is received when an attempt to use Auto Poll is made. If CU=2701 is coded with APL, a DTFBT MNOTE is given at assembly time. The Auto Poll feature is standard in the 2703. The switch operand must be coded SWITCH=NO.

Note: There is no special ordering of these codes in the sublist. Commas must be used to separate the different codes, but are not used to indicate missing ones.

Example: FEATURE=(APL,STC[,CHK])

BUFCB
Assigns a symbolic name to the buffer control block of the buffer pool used by the line group. Two or more line groups can share the same buffer pool if the DTFBT's for each specify the same name in their BUFCB operands.

BUFNO
Specifies the number of buffers to be provided in the buffer pool. Range: 2-255.

BUFL
Specifies the total length in bytes of each buffer in the pool. The length must be 12 bytes greater than the number of bytes required for data. Range: 16-32,760.

Note: If the keyword operands BUFNO, BUFL, and BUFCB are omitted, no buffer pool is created. However, BUFNO and BUFL can be omitted while BUFCB is coded, if BUFCB specifies the symbolic address of a buffer pool defined in a previous DTFBT in the same assembly by all three keywords.

SEPASMB
Defines how this DTFBT section will be assembled, where:

NO specifies this DTFBT will be assembled with the rest of the user's code (see Assembly Considerations).

YES specifies this DTFBT will be assembled separately. In this case a CATALR card with the name given to the DTFBT is generated ahead of the object deck (see Assembly Considerations).

MODNAME
Specifies the name of the BTAM logic module (BTMOD) which is to be link-edited with this DTFBT section. If MODNAME is omitted, the standard BTAM module name (IJLBTM) is used.

MSGL
This operand is used only for the 7770/7772 to specify the maximum input message length for any line in the line group when a WRITE invitational or WRITE conversational is issued. Range: 2-32767. The default option is 2.

TERMTST
Specifies whether the DTFBT table will contain the test activation code, where:

NO specifies the DTFBT will not contain the test activation code; therefore, no request for the on-line terminal test facility is made.

YES specifies the DTFBT will contain the test activation code. This is utilized by the on-line terminal test facility in determining if a test request has been made. YES may be specified only for IBM terminals, except for the 2260 Local and 7770/7772.

ERROPT
Specifies which ERP options (for non-binary synchronous transmission only) are to be performed where:

E specifies that error recovery is to be performed without text retry.

N specifies that no error recovery is to be performed.

R specifies that error recovery is to be performed with read text retry.

W specifies that error recovery is to be performed with write text retry.

RW specifies that error recovery is to be performed with read text retry and write text retry.

Note: This operand does not apply to Audio Response Units.

LERBADR

Specifies the symbolic address of the Line Error Block (LERB) associated with the line group.

MODELST

A sublist of codes is specified by the problem program (for use by BTAM at OPEN time) to define the following for each line in the line group (BSC only):

Checking mode - Normal or intermediate block checking (I.B.C.) mode may be selected. Only in I.B.C. mode is the Unit Separator (US) character recognized by the 2701 or 2703 as a control character.

Dual Communication Interface - Interface A or Interface B may be selected on the 2701 equipped with this feature.

Dual Code Feature - Code A or Code B may be selected on the 2701 equipped with the Dual Code Feature. The codes are:

	Meaning:		
Code	Checking	Interface	Code
0	Normal	A	A
1	Normal	A	B
2	Normal	B	A
3	Normal	B	B
4	I.B.C.	A	A
5	I.B.C.	A	B
6	I.B.C.	B	A
7	I.B.C.	B	B

For the 2703 or for the 2701 not equipped with the Dual Communication or Dual Code Features, a code should be used that specifies Interface A and Code A. If the MODELST operand is omitted, 0 is assumed for each line in the line group. If the MODELST operand is present, each line in the line group must be represented by a code unless 0 is intended, in which case only the trailing comma must be coded.

Example: If the line group is comprised of 5 lines, the user may code

```
MODELST=(0,4,,,) or
MODELST=(,2,,2,) etc.
```

A useful rule to remember is that the number of commas in the MODELST sublist must be equal to the number of commas in the LINELST sublist. The position of each code in the MODELST sublist corresponds to the relative line number of the line for which the code applies.

CTLCHAR

Specifies the transmission code used in binary synchronous communication for the line group. This operand is not used and is ignored for start-stop devices. It is mandatory for binary synchronous communication, since BTAM uses this operand to initialize a table of line control characters required for internal use by BTAM.

The following may be coded on the right side of the equal sign.

EBCDIC Specifies the EBCDIC line control characters are to be used.

ASCII Specifies the USASCII line control characters are to be used.

TRNSCD Specifies the 6-bit TRANSCODE line control characters are to be used.

RETRY

Specifies the number of retries by BTAM ERP for recoverable errors that occur on I/O operations for binary synchronous communication. The absolute expression specified on the right-side of the equal sign may range in value from 0 through 15. The operand may be omitted, in which case seven is assumed for binary synchronous communication.

Note: This operand is not applicable to non-binary synchronous devices.

CONFIG

Specifies the line configuration, where:

PPT specifies the data link between the CPU and the remote binary synchronous device is point-to-point.

MPT specifies the data link between the CPU and the remote binary synchronous devices is multipoint.

Name	Operation	Operands
symbol	DFTRMLST	listype- { OPENLST WRAPLST DIALST IDLST AUTOLST AUTOWLST SSALST SSAWLST } ,device dependent operands

Figure 5. Format of DFTRMLST Macro Instruction

This operand is not applicable to any BTAM-supported start-stop or local devices.

The following rules must be observed (see Assembly Considerations).

1. All DTFBT statements that are to share a buffer pool must be in the same assembly.

2. If SEPASMB=YES is specified in a DTFBT statement, normally there will be no other source code (except END) in the same assembly.

Note: The user must define the line group with a DTFBT macro instruction and must activate the line group with an OPEN macro before any message can be sent or received over a communication line of the line group.

DTFBTND (Define File End for BTAM) Macro Instruction

The DTFBTND macro instruction has no function in the present version of BTAM; it is included here to maintain compatibility with the previous version of BTAM. It is not necessary to specify DTFBTND because it no longer contains the CCW models for the various supported devices. The CCW models have been cataloged to the relocatable library and will be loaded at linkage-edit time.

Name	Operation	Operands
[symbol]	DTFBTND	[SEPASMB=NO SEPASMB=YES]

DFTRMLST (Define Terminal List) Macro Instruction

The DFTRMLST macro instruction (Figure 5) provides the facility for defining terminal lists.

symbol
Represents the symbolic name of the list. This symbol is mandatory because the name of the list is required in the various READ and WRITE initials issued in the user's program.

OPENLST
Specifies that the list to be constructed is an open polling list or an addressing list for a nonswitched line.

WRAPLST
Specifies that the list to be constructed is a wrap-around polling list for a nonswitched line.

DIALST
Specifies that the list to be constructed is for a line connection over a switched network.

IDLST
Specifies that the list to be constructed is for a line connection over a switched network requiring identification (ID) verification.

AUTOLST
Specifies that the list to be constructed is an open polling list used with the Auto Poll feature for binary synchronous communication.

AUTOWLST
Specifies that the list to be constructed is a wrap-around polling list used with the Auto Poll feature for binary synchronous communication.

SSALST
Specifies that the list to be

constructed is for an open start-stop Auto Poll list.

SSAWLST
Specifies that the list to be constructed is for a wrap-around start-stop Auto Poll list.

device dependent operands
Each list type requires different operands following the listype operand. These operands also can vary with the specific device involved. The section, Device Dependent Considerations, presents a detailed usage by the specific device of these operands. These operands provide control information to establish logical connection with the terminal.

Note: A maximum of 15 Read/Write operations can use one Auto Poll list because the Usage Count in the list keeps track of only 15 operations for CHGNTRY purposes. This applies to the following lists: SSALST and SSAWLST.

Line Error Block (LERB) Macro Instruction

The LERB macro instruction generates and initializes a table called the line error block, in which the Error Counts logic in BTMOD keeps by line the data check, intervention required, and non-text time-out error counts and the number of transmissions requested by the user (Figure 6). If any of the error counters reach their threshold values before the transmission count threshold is reached, a message will be printed on the System/360 Operator's Console. The message identifies the line and contains the three error counters, the transmission counter, and their respective threshold values.

Programming Note: Because the LERBADR operand of DTFBT generates a V-type address constant, the LERB macro may not appear in an unnamed control section.

After the message is printed, the transmission counter and the three error counters are added to a separate set of four cumulative counters, after which the counters are reset. Also, whenever the transmission counter reaches its threshold without any of the error counters reaching their thresholds, the four counters are added to the four cumulative counters and then the counters are reset.

<table>
<tr><th>Bytes</th><th colspan="4"></th></tr>
<tr><td>0</td><td colspan="4">Transmission Cumulative Counter</td></tr>
<tr><td>4</td><td colspan="2">Data Check Cumulative Counter</td><td colspan="2">Intervention Required Cumulative Counter</td></tr>
<tr><td>8</td><td colspan="2">Time Out Cumulative Counter</td><td>Transmissions Counter</td><td>Data Check Counter</td></tr>
<tr><td>12</td><td>Intervention Required Counter</td><td>Time Out Counter</td><td>Transmission Threshold</td><td>Data Check Threshold</td></tr>
<tr><td>16</td><td>Intervention Required Threshold</td><td>Time Out Threshold</td><td colspan="2">Reserved</td></tr>
</table>

Figure 6. Line Error Block

Name	Operation	Operands
symbol	LERB	nlines-absexp, [{([transmct-absexp], [datack-absexp], [intreq-absexp], [nontto-absexp]),}...]

nlines
Specifies the number of lines associated with this line group (DTFBT). 1 to 31 lines are allowed. This operand must be coded.

transmct
Specifies the number of consecutive transmission requests on a line that, when reached, causes the three error counters and the transmission counter to be added to their respective accumulators and the counters to be reset to zero. 1 to 255 transmissions are allowed; if this operand is omitted, a 255 transmission count is assumed.

datack
Specifies the threshold value for the number of data check errors for a specified number of transmissions on a line. A value of 1 to 255 is allowed, except that the value cannot be greater than the transmission count

(transmct). The default value is either 10 or the "transmct" value, whichever is less.

intreq
: Specifies the threshold value for the number of intervention required errors for a specified number of transmissions on a line. A value of 1 to 255 is allowed, except that the value cannot be greater than the "transmct". The default value is either 5 or the "transmct" value, whichever is less.

nontto
: Specifies the threshold value for the number of non-text time-out errors for a specified number of transmissions on a line. A value of 1 to 255 is allowed, except that the value cannot be greater than the "transmct". The default value is either 5 or the "transmct", whichever is less.

Only the first operand "nlines" is mandatory. If the first operand is the only one coded, the default threshold values of 255, 10, 5, and 5 will be supplied for all the lines specified by "nlines". If the threshold value is coded, it remains effective for any of the following lines for which a threshold value is not coded. It will remain effective until another set of threshold values is provided.

An example of the LERB macro instruction is as follows:

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

This coding will result in lines 0 and 1 having default thresholds of 255 "transmct", 10 "datack", 5 "intreq", and 5 "nontto"; lines 2, 3, and 4 having thresholds of (200,20,5,7) with 5 as the default value of "intreq"; and lines 5 through 9 having thresholds of (240,20,25,10). The parentheses are omitted if all the suboperands are omitted. The trailing commas are also omitted.

ASMTRTAB (Assemble Translation Table) Macro Instruction

The ASMTRTAB macro instruction provides the facility for assembling one or more DOS/BTAM translation tables with the problem program.

Name	Operation	Operand
Not used, ignored	ASMTRTAB	{tablename-code,}...

tablename
: Indicates the specified table (any "Table Name" listed in Figure 7 may be coded) is to be assembled with the problem program.

  Example: ASMTRTAB RSCI,SSCI

  This results in the assembling (in line) of the two translation tables: RSCI and SSCI, with the problem program.

Note: When the translation table(s) specified through ASMTRTAB is assembled, the name in the name field is of the form: IJLXXXX. The "XXXX" is the code (RSCI, SSCI, etc.) appearing in the operand field of the ASMTRTAB macro. This IJLXXXX name would need to be coded by the user as the symbolic address of the table in any "LA," "MVC", or "TR" instruction of his own, if he intends to use or manipulate the table independent of BTAM. However, only the "XXXX" code should appear in the operand field of the TRNSLATE macro (see that section). The user is cautioned that user-defined translation tables should not be given names identical to those listed in Figure 7.

Note 1. The RF50 (OR RF40) translation table converts lowercase alphabetic characters from 1050 (or 2740) code to their uppercase EBCDIC equivalents. The RC50 (OR RC40) translation table converts lowercase 1050 (or 2740) code to lowercase EBCDIC.

Note 2. The RASA and SASA translation tables are used for binary synchronous communication.

Note 3. The RSCI and SSCI translation tables are used with the 2848 when it is attached to a 2701 via IBM Terminal Control Type III.

TRANSIENT MACROS

The OPEN, LOPEN, and CLOSE phases are executed in the DOS transient area.

Table Name		Type of Conversion	Type of Terminal
For incoming messages:			
RC30		1030 Code to EBCDIC	IBM 1030
RC60		1060 Code to EBCDIC	IBM 1060
RC50	Note 1	1050 Code to EBCDIC	IBM 1050
RC40	Note 1	2740 Code to EBCDIC	IBM 2740
RF50	Note 1	1050 Code to EBCDIC	IBM 1050
RF40	Note 1	2740 Code to EBCDIC	IBM 2740
RASA	Note 2	USASCII Code to EBCDIC	Binary Synchronous
RSCI	Note 3	USASCII Code to EBCDIC	IBM 2848
RCT1		Baudot Code to EBCDIC	AT&T 83B3,WU115A
RCT2		TWX Code to EBCDIC	AT&T 33,35
RC80		6-Bit TRANSCODE to EBCDIC	IBM 2780
For outgoing messages:			
SD30		EBCDIC to 1030 Code	IBM 1030
SD60		EBCDIC to 1060 Code	IBM 1060
SD50		EBCDIC to 1050 Code	IBM 1050
SD40		EBCDIC to 2740 Code	IBM 2740
SASA	Note 2	EBCDIC to USASCII Code	Binary Synchronous
SSCI	Note 3	EBCDIC to USASCII Code	IBM 2848
SCT1		EBCDIC to Baudot Code	AT&T 83B3,WU115A
SCT2		EBCDIC to TWX Code	AT&T 33,35
SD80		EBCDIC to 6-Bit TRANSCODE	IBM 2780

Figure 7. Code Translation Tables Provided by BTAM

OPEN Macro Instruction

The OPEN macro prepares communication line groups for use and executes, when applicable, any required channel programs.

Name	Operation	Operands
[symbol]	OPEN	$\left\{\begin{matrix}\text{dtfname-symbol,}\\\text{(r),}\end{matrix}\right\}$...

dtfname
Specifies the symbolic name of the DTFBT macro instruction associated with the line group to be opened. Several line groups may be opened with a single OPEN macro by entering their DTFBT names as operands. If more than one DTFBT is specified, both symbols and register notation may appear in the operand field. For example, if the user wishes to open DTFBT1, DTFBT2, DTFBT3, and DTFBT4, he might code:

```
STEP1 OPEN DTFBT1,(2),(3),DTFB4
```

In this case, register 2 would have been loaded with the address of DTFBT2 and register 3 with the address of DTFBT3 prior to issuing the OPEN macro. This macro instruction can be used to open other types of files. As many as 16 files may be specified in a single OPEN macro instruction.

A line group must be opened before any message can be sent or received over any line in the group and before buffers from the associated buffer pool are obtained for that line group. In certain cases there is a line delay before transmission can occur on the line. In such cases, a message indicating this line delay is written on the 1052 Printer.

For the 2260 Local, the OPEN macro erases the screens of all the Display Stations specified in the LINELST operand of the DTFBT.

LOPEN Macro Instruction

The LOPEN macro instruction is used to condition a line on which an error occurred during OPEN, and, when necessary, initiates line delays.

Name	Operation	Operand
[symbol]	LOPEN	decb-symbol (r)

decb
Specifies the symbolic name of the DECB associated with the line to be opened. Only one line may be conditioned with a single LOPEN macro instruction. Register notation may also appear in the operand field.

If the user issues a READ, WRITE, or CONTROL macro instruction on a line that failed to open, a return code of X'14' is given to him in register 15. He may then issue an LOPEN macro instruction to condition the line. If LOPEN successfully conditions the line, completion code X'7F' is posted; otherwise the completion code X'41' is posted and the line should be regarded as temporarily unusable.

CLOSE Macro Instruction

The CLOSE macro instruction removes communication line groups from use. If necessary, active channel programs will be halted.

Name	Operation	Operands
[symbol]	CLOSE	dtfname-symbol, ... (r),

dtfname
Specifies the symbolic name of the DTFBT macro instructions associated with the line group to be closed. Several line groups may be closed with a single CLOSE macro by entering their DTFBT names as operands. In addition, both symbols and register notation may be mixed in the operand field of the macro if more than one DTFBT is specified. This macro may also be used to close other types of files. As many as 16 files may be specified in a single CLOSE macro instruction.

The CLOSE macro instruction is used with the OPEN macro instruction to control the operative status of communication lines in the system.

DATA HANDLING MACROS

The data handling macro instructions are the following:

CONTROL

READ

WRITE

CHGNTRY

RESETPL

WAIT

TWAIT

LERPRT

TRNSLATE

Creation of a DECB (Data Event Control Block)

The READ, WRITE, and CONTROL macro instructions have different functions, depending on how the MF keyword operand is coded:

1. MF=L: a DECB is reserved at assembly time and the parameters coded in the macro instruction initialize the DECB. Because this is an assembly time function, no register notation is allowed. The function (READ, WRITE, or CONTROL) is not executed. This form of the macro should not be branched to by executable code.

2. MF=E: the parameters coded in the macro instruction update the fields of a DECB previously defined, and the function (READ, WRITE, or CONTROL) is executed. Register notation is allowed where indicated.

3. MF operand omitted: a DECB is reserved and initialized at assembly time, and the function (READ, WRITE, or CONTROL) is executed at execution time. With the exception of the decb operand, register notation may be used where indicated, and the operands so coded update the corresponding DECB fields prior to execution. If a READ, WRITE, or CONTROL of the MF=E form is issued using a DECB generated by a macro with the MF operand omitted, the values in the DECB generated at assembly time are replaced by those values specified in the MF=E form of the macro.

CONTROL Macro Instruction

The CONTROL macro instruction (Figure 8) is used for the 7770/7772 to:

1. Enable a line to receive a call from a remote device. When connection is established, the operation is posted complete.

2. Disconnect a line after the transmission is completed.

For binary synchronous communication, the CONTROL macro can be used for function 2 (above) for switched connections and/or to:

3. Set the checking mode of the binary synchronous adapter of the attached control unit, regardless of the line connection (switched or nonswitched).

decb
: Specifies the address of the DECB associated with the line. The user is permitted to load a specified register with the address of the DECB prior to issuing CONTROL only when MF=E is coded.

optype
: Specifies which CONTROL function is to be performed, where:

  TI specifies CONTROL Initial. Tne line is to be enabled to receive a call (7770/7772 only).

  TD specifies CONTROL Disable. The line is to be disconnected because the answering of a successfully received call is ended and no more transmission is to occur. If the MODE operand is present, the function of CONTROL Mode (TM) is also to be executed.

  TM specifies CONTROL Mode. This optype is used for binary synchronous communication only. The "mode byte" in the DECB is to be posted and the mode of the control unit adapter is to be set. Note: A CONTROL TM need be issued only if the checking mode is to be changed from that specified via the MODELST operand in DTFBT.

dtfbt
: Specifies the address of the DTFBT table for the line group. Register notation permits the user to load the specified register with the address of the DTFBT prior to execution.

area, length, entry
: These operands need only be used when the CONTROL macro instruction is used to create a DECB that will be subsequently used for READ or WRITE operations. Execution of the CONTROL macro does not involve the use of these operands, however, because no transfer of data is performed.

rln
: Specifies the relative line number (0-30) within the line group.

MODE
: Specifies how the checking mode of the binary synchronous adapter of the attached control unit is to be set. This operand must not be used for start-stop transmission.

  0 specifies normal checking mode is to be set. In this mode the 2701 or 2703 does not recognize the Unit Separator (US) character as a control character. If the MODE operand is omitted in the CONTROL TM form, 0 is assumed.

```
┌─────────┬──────────┬──────────────────────────────────────────┐
│Name     │Operation │ Operands                                 │
├─────────┼──────────┼──────────────────────────────────────────┤
│[symbol] │CONTROL   │ ⎧decb-symbol⎫          ⎧TI⎫              │
│         │          │ ⎨    (r)    ⎬,optype-  ⎨TD⎬              │
│         │          │ ⎩    (1)    ⎭          ⎩TM⎭,             │
│         │          │                                          │
│         │          │ ⎡dtfbt-symbol⎤ ⎡area-relexp⎤             │
│         │          │ ⎣    (r)     ⎦,⎣    (r)    ⎦,            │
│         │          │                                          │
│         │          │ ⎡length-absexp⎤ ⎡entry-relexp⎤           │
│         │          │ ⎣     (r)     ⎦,⎣    (r)     ⎦,          │
│         │          │                                          │
│         │          │                 ⎡      ⎧ 0 ⎫⎤⎡    ⎧L⎫⎤   │
│         │          │ ⎡rln-absexp⎤    ⎢,MODE=⎨ 1 ⎬⎥⎢,MF=⎨ ⎬⎥   │
│         │          │ ⎣   (r)    ⎦    ⎣      ⎩(r)⎭⎦⎣    ⎩E⎭⎦   │
└─────────┴──────────┴──────────────────────────────────────────┘
```

Figure 8. Format of CONTROL Macro Instruction

1 specifies intermediate block checking mode is to be set. In this mode the 2701 or 2703 recognizes the US character as providing the end-of-intermediate-block function (in other IBM publications on the data link control procedures for BSC this function is called ITB - the character used in all codes for this function is the US, however). Essentially, this checking mode enables a single transmission to be sub-divided into individually checked blocks without line-turnaround at the end of each intermediate block. On a receive operation the 2701 or 2703 sends an error information byte (E.I.B.) into main storage following each US, ETB, or ETX received from the line. (See General Information in the 2780 device dependent section for the format of the E.I.B.) In this mode US is provided by the user as the end character for each intermediate block, but ETB or ETX must be inserted by the user as the last end character for the total transmission (so that line-turnaround can occur).

BTAM support for this checking mode does not include analysis of E.I.B.'s and precludes use of DLE US on transparent write operations (although the intermediate block checking mode can still be set). BTAM provides a READ Repeat with Leading Graphics for the user who desires to analyze E.I.B.'s and request only partial retransmission (beginning with the first block in error) of a message from a remote CPU. Through this macro the user can send leading graphics with NAK identifying the first block received in error. When this macro is used BTAM ERP will not retry data check and overrun errors on a received message (received on the Read command in the READ TPL channel program). Similarly, when leading graphics with NAK are received (via a conversational-WRITE) BTAM ERP will not automatically retransmit the previously sent message. Instead, the user is given control. In all other cases BTAM performs error correction through automatic total retransmission. For the 2780, partial retransmission is not possible since I.B.C is hardware-implemented in a different manner.

(r) 0 or 1 is loaded in the low-order byte of the register.

Note: The MODE keyword operand can be used with optypes TD and TM. If TD is specified the line will be disconnected (disabled). The mode is then set, only if the MODE operand is coded. Whenever the MODE operand is coded the "mode byte" in the DECB is updated. Thus, although the MODE operand is required for the TD optype if the mode is to be set, it may be omitted when TM is specified, provided the "mode byte" in the DECB was initialized by a previously issued CONTROL.

MF

In addition to the operations performed as previously explained (See Creation of a DECB), the coding of this operand results in the following actions.

MF=L If the MODE operand is present the mode byte in the DECB is initialized. Register notation is not used.

MF=E If optype TD or TM is specified and the MODE operand is present, the code (or low-order byte in the register) is used to set the mode of the adapter. If optype TM is specified and the MODE operand is not present, the previously posted mode byte in the DECB is used to set the mode of the adapter.

No MF The MODE operand must be present if the mode of the adapter is to be set.

READ Macro Instruction

A READ macro instruction (Figure 9) is issued by the problem program whenever transmission from a device is required and/or the user desires to reserve and initialize a DECB. Various options are permitted, not all of which apply to every device supported. Figure 10 lists the options available for each device. In the section Device Dependent Considerations, further information on specific implementation of the READ options is detailed for each supported device.

decb

Specifies the address of the DECB associated with the line. The user is permitted to load a specified register with the address of the DECB prior to issuing a READ only when MF=E is used.

optype
: Specifies the specific READ option (TI, TT, TIR, etc., see Figure 10.)

dtfbt
: Specifies the address of the DTFBT table for the line group.

area
: Specifies the address of the first byte of the input area used to receive the message. If 'S' (apostrophes are mandatory) is coded, BTAM reads the message into the buffer(s) of the buffer pool associated with the line group. For the 2260 Local only one buffer will be supplied.

<u>Note</u>: On completion of a READ operation with area 'S' specified, the problem program should check bit 6 of the Flag byte to see if there are unreleased buffers not used. If this bit is on, the last 3 bytes of the ECB contain the address of the first buffer not used by the problem program in the READ operation. The problem program should release the unused buffer(s) to the buffer pool with a RELBUF macro.

length
: Specifies the maximum number of bytes in the expected message. (This operand is used to determine the maximum number of buffers needed when area is coded 'S'. Unused buffers are automatically returned to the buffer pool if the READ completes normally. If an error is posted on a READ operation using buffers, the user must return the acquired buffers by a RELBUF macro instruction before retrying the operation.)

entry
: Specifies for:

1. A nonswitched line (OPENLST, WRAPLST, SSALST, or SSAWLST), the address of an entry within the terminal list.
2. A switched line (DIALST), the address of the beginning of the terminal list at initial contact time and an entry afterwards.
3. Binary synchronous applications (DIALST, AUTOLST, AUTOWLST, and IDLST) or Audio units, an appropriate address as specified for the specific remote device in the section, <u>Device Dependent Considerations</u>. The 'S' option has a special use (see Figure 22) in READ Continue for CPU-to-CPU Dial. If <u>relexp</u> or (r) is coded, the address specified is placed in the "Entry Field" of the DECB and also in the "Polling Pointer" of the DECB.

   The current polling pointer will be moved into the entry field of the DECB (See <u>Data Event Control Block</u>) if 'S' is coded.

   <u>Note</u>: The entry operand is not used for the 2260 Local. Therefore, only the consecutive commas should be coded.

rln
: Specifies the relative line number within the line group. (Range 0-30 inclusive.) Details of the specialized use of this operand for the 2260 Local are discussed in the device dependent section for that device. At the completion of a READ for the 2260 Local, the rln for the display station from which data has been read is placed in the low-order byte of the DECB polling pointer field.

MF
: See <u>Creation of a DECB</u>.

```
┌─────────┬───────────┬──────────────────────────────────────────────────┐
│ Name    │ Operation │ Operands                                         │
├─────────┼───────────┼──────────────────────────────────────────────────┤
│[symbol] │   READ    │ ⎧decb-symbol⎫                  ⎡dtfbt-symbol⎤   │
│         │           │ ⎨    (r)    ⎬ ,optype-code,    ⎣     (r)    ⎦,  │
│         │           │ ⎩    (1)    ⎭                                   │
│         │           │                                                  │
│         │           │ ⎡area-relexp⎤ ⎡length-absexp⎤                    │
│         │           │ ⎢    (r)    ⎥,⎣     (r)     ⎦,                   │
│         │           │ ⎣    'S'    ⎦                                    │
│         │           │                                                  │
│         │           │ ⎡entry-relexp⎤                                   │
│         │           │ ⎢    (r)     ⎥, ⎡rln-absexp⎤ ⎡,MF=⎧L⎫⎤            │
│         │           │ ⎣    'S'     ⎦  ⎣(r)       ⎦ ⎣    ⎩E⎭⎦            │
└─────────┴───────────┴──────────────────────────────────────────────────┘
```

Figure 9. Format of READ Macro Instruction

OPTION	TYPE CODE	1030	1050	Switch 1050	1060	Remote 2260	Local 2260	83B3 115A	7770 7772	TWX 33/35	2740*
READ Initial	TI	X	X	X	X	X	X	X	X	X	1
READ Initial with Reset	TIR	X	X	X	X					X	2
READ Initial with Lock	TIL						X				
READ Continue	TT	X	X	X	X	X					3
READ Continue with Reset	TTR	X	X	X	X						3
READ Conversational	TV			X						X	4
READ Conversational with Reset	TVR			X						X	4
READ Repeat	TP	X	X	X	X	X					3
READ Repeat with Reset	TPR	X	X	X	X						3
READ Buffer	TB					X	X				
READ Buffer with Lock	TBL						X				
READ Skip	TS	X	X	X	X	X		X		X	1
READ Short	TO						X				
READ Short with Lock	TOL						X				
WRITE Initial	TI	X	X	X	X	X	X	X	X	X	1
WRITE Initial with Reset	TIR	X	X	X	X	X		X	X	X	1
WRITE Initial with Lock	TIL						X				
WRITE Initial Optical	TIO										6
WRITE Invitational	TC								X		
WRITE Invitational Optical	TCO										6
WRITE Continue	TT	X	X	X		X			X		3
WRITE Continue with Reset	TTR	X	X	X		X			X		3
WRITE Conversational	TV			X					X	X	5
WRITE Conversational with Reset	TVR			X						X	5
WRITE at Line Address	TL					X	X				
WRITE at Line Address with Reset	TLR					X					
WRITE at Line Address with Lock	TLL						X				
WRITE Erase	TS					X	X				
WRITE Erase with Reset	TSR					X					
WRITE Erase with Lock	TSL						X				
WRITE Break	TB							X			
WRITE Positive Acknowledge	TA	X	X	X	X	X					3
WRITE Negative Acknowledge	TN	X	X	X	X					X	2
CONTROL Initial	TI								X		
CONTROL Disable	TD								X		

1 - All 2740
2 - 2740 with C, D, DC, DT, DTC, SC
3 - With C, DC, DTC, SC
4 - With D, DC, DT, DTC
5 - With C, D, DC, DT, DTC
6 - With C and 2760 OIU, or with D, C, and 2760 OIU

Figure 10. READ/WRITE/CONTROL Options (Part 1 of 2)

OPTION	TYPE CODE	CPU-CPU Contention	CPU-CPU Dial
Read Initial	TI	X	X
Read Initial Inquiry	TIQ	X	
Read Continue	TT	X	X
Read Continue with Leading Graphics	TTL	X	X
Read Repeat	TP	X	X
Read Repeat with Leading Graphics	TPL	X	X
Read Inquiry	TQ	X	X
Write Initial	TI	X	X
Write Initial Transparent Block	TIE	X	X
Write Initial Transparent Text	TIX	X	X
Write Initial Transparent Conversational	TIXV	X	X
Write Initial Conversational	TIV	X	X
Write Continue	TT	X	X
Write Transparent Block	TE	X	X
Write Transparent Text	TX	X	X
Write Transparent Conversational	TXV	X	X
Write Conversational	TV	X	X
Write Disconnect	TD		X
Write End-of-Transmission	TR	X	X
Write Wait-Before-Transmitting (WABT)	TW	X	X
Write Inquiry	TQ	X	X
Control Disable	TD		X
Control Mode	TM	X	X

Figure 10. READ/WRITE/CONTROL Options (Part 2 of 2)

WRITE Macro Instruction

A WRITE macro instruction (Figure 11) is issued by the problem program whenever transmission to a device is required and/or the user wishes to reserve and initialize a DECB. Figure 10 lists the various WRITE options. Specific implementation of these options for each device is discussed in the section Device Dependent Considerations.

```
┌──────────┬──────────┬──────────────────────────────────────────────┐
│Name      │Operation │ Operands                                     │
├──────────┼──────────┼──────────────────────────────────────────────┤
│[symbol]  │WRITE     │┌decb-symbol┐              ┌dtfbt-symbol┐    │
│          │          ││    (r)    │,optype-code, └    (r)     ┘,   │
│          │          │└    (1)    ┘                                 │
│          │          │                                              │
│          │          │┌area-relexp┐ ┌length-absexp┐                 │
│          │          │└    (r)    ┘,│     (r)     │,                │
│          │          │              └     'S'     ┘                 │
│          │          │                                              │
│          │          │┌entry-relexp┐ ┌rln-absexp┐ ┌,MF=┌L┐┐         │
│          │          ││     (r)    │,└    (r)   ┘ └    └E┘┘         │
│          │          │└     'S'    ┘                                │
└──────────┴──────────┴──────────────────────────────────────────────┘
```

Figure 11. Format of WRITE Macro Instruction

decb
Specifies the address of the DECB associated with the line. The user is permitted to load a specified register with the address of the DECB prior to issuing a WRITE when MF=E is coded.

optype
Specifies the option (TI, TT, T1R, etc, see Figure 10).

dtfbt
Specifies the address of the DTFBT table for the line group.

area
Specifies the address of the first byte of the output area used to send the message or the address of the first buffer in a chain of buffers. For WRITE TCO, this operand is the address of the first byte of the input area for the invited response from the 2760 Optical Image Unit. This operand is ignored for WRITE TIO.

length
Specifies the number of bytes in the outgoing message. If this operand is coded 'S' (apostrophes are mandatory), BTAM is to write a chain of buffers. The address of the first buffer in this case is specified in the area operand. It is the responsibility of the user to chain the buffers forward if the normal BTAM chaining is altered. When 'S' is coded, BTAM generates the channel program in the buffers rather than the LCP (except 2260 Local). One Write CCW is set up in each buffer to write out the contents of the buffer. BTAM places in the count field of each Write CCW a value equal to the buffer size minus 12. Then, by use of data-chaining and TIC commands, the contents of the full chain of buffers are transmitted. Normally, an end character appearing in the last buffer will terminate the write operation even if the count in the last Write command has not been exhausted.

A special technique must be employed for those cases when no end character appears in the last buffer and the user does not wish to transmit the full contents of the last buffer in the chain (not applicable to 2260 Local, however). After getting the required number of buffers via REQBUF, the user should place in bytes 6-7 (the 7th and 8th physical bytes) of the last buffer in the chain the exact count of data characters to be transmitted from this buffer. This should be done by a Store Halfword (STH) instruction to avoid modifying the neighboring bytes, which are initialized by BTAM. After adjusting the count in the last buffer, the user can then issue a WRITE macro with length coded as 'S'. This technique is useful when writing from buffers using one of the BSC transparent-WRITE macros. It is also convenient when writing from buffers to a TWX 33/35 (no end character is used). The same technique can be employed for a message switching application where the user wants to write out from buffers just filled from a previous READ macro. (See the section, Buffering, for details.) For the 2740 and 2260 Local the message must be contained entirely within one buffer.

For WRITE TCO, this operand is the length of the input area for the invited response from the 2760 Optical Image Unit. This operand is ignored for WRITE TIO.

entry
Specifies for:
1. Audio Response Units (this operand will be used only for WRITE Invitational and WRITE

conversational), the address of the first byte of the input area. In that case, the length of the input area will be taken from the MSGL operand in the DTFBT macro instruction.

2. A nonswitched line, the address of an entry in the terminal list. For the 2760 Optical Image Unit on a 2740C, the address of the first of the user's three frame-change control characters, which may be set up by assembler language, DFTRMLST OPENLST, or execution.
3. A switched line, the address of the beginning of the terminal list at initial contact time and of an entry afterwards. The address specified is placed in the entry field of the DECB. For the 2760 Optical Image Unit on a 2740DC, the address of a DFTRMLST DIALST specifying as the list-entry (following the dial characters) the three frame-change characters, when the optype is TIO. On WRITE TCO, the address of the first of the frame-change characters.
4. Binary synchronous applications, an address as specified for the particular device in the Device Dependent Considerations section. Special usage is made of this operand by the Conversational-type WRITE's. In addition, the 'S' form may be used for retrying an ID Verification procedure when an error occurs, without redialing (see CPU-to-CPU Dial). The address specified is placed in the "Entry Field" of the DECB. For the binary synchronous support it is also placed in the "Address Pointer" of the DECB.

Note: The entry operand is not used for the 2260 Local, therefore two consecutive commas should be coded.

rln

Specifies the relative line number within the line group. (Range: 0-30 inclusive.) For the 2260 Local, this operand specifies the relative position in the DTFBT of the LCB for the display station or printer to which BTAM will write.

MF

See Creation of a DECB.

Programming Notes

Execution of a READ, WRITE, or CONTROL macro instruction causes control to be passed to a BTAM routine that constructs channel programs. If no invalid conditions are detected, a channel program will be generated for the requested I/O operation, and control will be passed to the DOS supervisor for execution. Once the channel program has been started, control will be returned to the user's problem program with a return code of zero in register 15 and the condition code set to zero.

The I/O operation proceeds asynchronously with respect to the user's problem program execution. When the problem program wishes to continue processing only after the I/O operation has completed, a WAIT or TWAIT macro instruction may be issued.

If, as a result of a READ, WRITE, or CONTROL macro instruction, control is returned to the problem program with a nonzero return code, no I/O operation was started. Therefore, the problem program must not issue a WAIT or TWAIT macro instruction (the program would enter a permanent wait state).

If the problem programmer intends to make use of the 'S' option for the entry operand in the READ Initial (TI) macro instruction, he must ensure that the polling list address is placed in the polling pointer field of the data event control block before the first execution of the READ Initial (TI) macro instruction. Once the polling pointer has been initialized, the 'S' option for the entry operand may be used freely because the polling pointer is preserved across write operations. Figures 12 and 13 show how various conditions affect the entry field, polling pointer, and address pointer in the DECB.

A Read/Write sequence on a switched network must always be terminated by a macro instruction which disconnects the line. This may be a Read or Write macro instruction specified with the reset option, a WRITE Positive Acknowledge (TA), or a WRITE Negative Acknowledge (TN).

The switched network user should issue a WRITE TA or a WRITE TN at the beginning of his program to reset the line.

When a READ or WRITE macro instruction specifies "reset at completion" (TIR, TTR, TPR, or TVR) for a switched-connection line, BTAM disconnects the line only if no error condition occurs during the execution of the basic channel program. Thus, the problem program may attempt retransmission without re-establishing the line connection. If the problem program elects not to attempt retransmission, the WRITE Negative Acknowledge macro instruction (optype: TN) may be executed to perform the disconnect function.

Situation	Entry Field	Polling Pointer
READ macro expansion: entry operand coded as relexp or (r).	Absolute address of the entry specified by the macro.	Absolute address of the entry specified by the macro.
READ macro expansion: entry operand coded as 'S'.	The address in the polling pointer field of the DECB is moved into the entry field.	Unchanged
Positive response to polling.	Address of the entry containing the polling characters of the sending device.	Address of the entry containing the polling characters of the sending device.
Negative response to polling after a RESETPL macro has terminated further polling.	Address of the entry containing the polling Characters of the last polled device.	Address of the first nonskipped (active) entry following the entry containing the polling characters of the last polled device. If no active entry follows, the address of the beginning of the list is used instead.
Negative response to polling after the end of an open list has been reached.	Address of the entry containing the polling characters of the last polled device.	Address of the beginning of the list.

Figure 12. Contents of the Entry Field and Polling Pointer

Situation	Entry Field	Address Pointer
WRITE macro expansion	Absolute value of the symbolic address specified in the macro.	This field is ignored.
Negative response to addressing	Address of the entry	Address of the entry
Negative response to multiaddressing	Address of the first entry in the list.	Address of the entry that contains the addressing characters of the device that sent the negative response.
Positive response to addressing/multiaddressing	Address of the first entry in the list.	Address of the last entry in the list.

Figure 13. Contents of the Entry Field and Address Pointer

When a READ or WRITE macro instruction is specified with the "Lock Option" (TIL, TOL, TBL, TLL, or TSL) for a 2260 Local, the manual entry keyboard retains the same locked or unlocked status that existed at the beginning of the execution of the oper ation requested by the macro instruction. If the keyboard is unlocked prior to the execution, the keyboard will be unlocked a the completion of the operation. A keyboar that is locked prior to the execution will remain locked at the completion; this prevents further operator entry until the ERASE key or the RESTORE key is depressed or until a BTAM macro instruction without the Lock option is issued.

EOT is recognized by the start-stop control unit (2701, 2702, 2703) and sets the control unit in control mode, but the Write command is not terminated. The Write command terminates when the residual count reaches zero. If the characters are not part of the character set recognized by the terminal adapter (2701) or the terminal control (2702, 2703), unit check with data check is set in the sense byte. For instance, X'00' will bring this condition. Therefore, it is safe to set the buffer area to a printable character for the device at the beginning of the problem program.

When a WRITE macro instruction specifies "reset at completion" for a nonswitched line, the EOT sequence will be transmitted only if no error condition occurs during execution of the basic channel program.

When a READ macro instruction involving polling results in a message being received, the polling pointer in the DECB contains the address of the entry in the polling list that was last polled.

Note: The polling pointer in the DECB is not updated upon receiving a positive response to polling using an Auto Poll list. In this case the polling pointer does not contain the address of the entry in the polling list that was last polled.

Thus, the problem program may determine the source of the message by inspecting the contents of the polling list entry at that address except when Auto Poll is used. Note that the polling pointer in the DECB is not modified by addressing operations. If the 'S' option for entry is coded in the next READ to be executed, polling proceeds with the device last polled. Thus, by following a successful READ macro with another READ using the 'S' option for the entry operand, the user accomplishes an exhaustive polling of a device before proceeding to poll the next device listed as an entry in the same terminal list.

CHGNTRY (Change Terminal Entry) Macro Instruction

The CHGNTRY macro instruction (Figure 14) provides the means for skipping (deactivating) or reactivating a terminal entry in a list. The user must not issue CHGNTRY for SSALST or SSAWLST while the list is being polled via READ initial (see Return Codes). No such constraint exists for the other listypes. However, the user is cautioned not to skip the only remaining active entry in an AUTOWLST during a READ initial that is currently using the list.

list
: Specifies the symbolic name of the DFTRMLST macro instruction that defined the terminal list containing the entry to be skipped or activated. If (r), any register 2-12, is coded, the user must have loaded the address of the terminal list into the register specified.

Name	Operation	Operands
[symbol]	CHGNTRY	$\left\{\begin{matrix}\text{list-symbol}\\ \text{(r)}\end{matrix}\right\}$, listtype-$\left\{\begin{matrix}\text{OPENLST}\\ \text{WRAPLST}\\ \text{DIALST}\\ \text{AUTOLST}\\ \text{AUTOWLST}\\ \text{SSALST}\\ \text{SSAWLST}\end{matrix}\right\}$ $\left\{\begin{matrix}\text{position-absexp}\\ \text{(r)}\end{matrix}\right\}$,$\left\{\begin{matrix}\text{numchars-absexp}\\ \text{(r)}\end{matrix}\right\}$,action-$\left\{\begin{matrix}\text{SKIP}\\ \text{ACTIVATE}\end{matrix}\right\}$

Figure 14. Format of CHGNTRY Macro Instruction

listype
Specifies the type of list specified in the DFTRMLST macro that defined the terminal list. The CHGNTRY macro is not used for the type, IDLST, so IDLST would never be coded here.

position
Specifies the position of the entry in the specified terminal list. The first entry in the list is in position 0, the second is in position 1, and so forth. The maximum position that can be specified is a function of the listype: OPENLST, WRAPLST, DIALST - 31; AUTOLST, AUTOWLST - 97; SSALST, SSAWLST - 98.

numchars
Specifies the number of polling or addressing characters in the specified entry. (The number does not include the control byte provided by BTAM for each entry or the "SYN" byte provided for each AUTOLST/AUTOWLST entry - See Appendix A.) For SSALST and SSAWLST, this operand is omitted and only the comma is coded.

action
Specifies the operation to be performed on the specified entry, where:

SKIP indicates the entry is to be skipped when polling or addressing with the list. This means the skip (S) bit is turned ON in the control byte for the entry in the case of OPENLST, WRAPLST, and DIALST. For AUTOLST and AUTOWLST (binary synchronous Auto Poll lists), the ENQ character and the control (index) byte for the entry are replaced by EOT and SYN characters (see Appendix A). For SSALST and SSAWLST (start-stop Auto Poll open list and wrap-around list) the list is physically restructured, with the entry to be skipped extracted from the list and saved at the end of the list. The value in the index byte is unchanged.

ACTIVATE indicates the entry is to be reactivated. This is done by turning OFF the skip (S) bit in the control byte for the entry in the case of OPENLST, WRAPLST, and DIALST. For AUTOLST and AUTOWLST, the EOT, SYN characters are replaced by the original ENQ character and control byte for the entry.

For SSALST and SSAWLST the list is physically restructured, with the entry to be activated extracted from the previously skipped entries saved at the end of the list and merged into the active entries by index order. The value in the index byte is unchanged.

Shown below is the special form of CHGNTRY for the 2260 Local.

Name	Operation	Operands
[symbol]	CHGNTRY	{dtfbt-symbol (r)},ATTLST, {position-absexp (r)}, action-{SKIP ACTIVATE}

dtfbt
Specifies the symbolic address of the DTFBT defining the 2260 Local line group.

ATTLST
Specifies that the line control blocks (LCB's) of the DTFBT are to be treated logically as an "attention list" for 2260 Local.

position
Specifies the relative position (0-24) within the DTFBT of the LCB for the display station to be skipped or activated on a READ initial or READ short operation.

action
Specifies the action to be performed:

SKIP denotes that BTAM is to ignore the 2260 corresponding to the LCB.

ACTIVATE denotes that BTAM is to resume processing the 2260 associated with the LCB.

RESETPL Macro Instruction

The RESETPL macro instruction provides the problem program with the facility to:

1. Interrupt a polling sequence.

2. Terminate a Prepare command (BSC).

3. Free a line for which an Enable command was issued but not completed (no call was received).

4. Terminate a wrap-around Auto Poll sequence.

5. Free for other operations a 2260 Local DTFBT for which a READ Initial or READ Short has been requested through the indicated DECB, but for which no response has been received.

Name	Operation	Operands
[symbol]	RESETPL	decb-symbol (r) (1)

decb
Specifies the address of the DECB associated with the line.

WAIT Macro Instruction

The WAIT macro instruction relinquishes control of the CPU when the problem program has no further processing to do and must wait for the completion of one or more events.

Note: The user is cautioned about issuing a WAIT for a channel program which includes channel commands that do not time-out. Commands such as Enable, Prepare and Poll (when polling terminals which have nothing to send and a wrap-around list is being used) might not complete in a reasonable time period.

count
Specifies the number of events among the events referred to by the ECB or ECBLIST operand that must be posted complete before the WAIT macro is satisfied. If the count operand is omitted, 1 is assumed.

Name	Operation	Operands
[symbol]	WAIT	[count-absexp, (r), (0),] ECB= symbol / (r) / (1) ECBLIST= relexp / (r) / (1)

ECB
Specifies the address of an Event Control Block (ECB) representing a single event to be posted complete before processing by the problem program can continue. (An ECB consists of the first four bytes of a DECB.)

ECBLIST
Specifies the address of a variable-length list containing fullword entries, with each fullword entry containing the address of an event control block in the low-order three bytes. Each event control block pointed to represents an event awaiting completion. In this list of ECB addresses the high order bit (0-bit) of each fullword entry except the last in the list must be zero. In the last entry in the list, the user must set the 0-bit in the high-order byte of the entry to one.

TWAIT Macro Instruction

The TWAIT macro instruction relinquishes control of the CPU when the problem program must wait for the completion of one of a number of events before further processing can be done. (See the Note for WAIT.)

Name	Operation	Operands
[symbol]	TWAIT	(r),[TERMTST,] ECBLIST= relexp / (r) / (1)

(r)
This user-specified register contains the address of the ECB posted as complete when the TWAIT is satisfied. (An ECB consists of the first four bytes of a DECB.)

TERMTST
: Specifies that the on-line terminal test facility will be provided. If this operand is omitted, the facility will not be provided. If this operand is coded and the terminal test logic is not included in the BTAM module, a return code of 04 is placed in register 15.

  If TWAIT is specified for a device which is not supported by the TERMTST facility, a test request for that device will be ignored.

ECBLIST
: Specifies the address of the user-created list of ECB addresses representing events awaiting completion. Each entry in the list is a fullword containing an address in the low-order three bytes. Because the list is of variable length, the high-order bit (0-bit) of each fullword entry must be set to zero except that of the last entry. The high-order bit of the last fullword entry must be set to one to identify the entry as the last in the list.

Note: If the problem program issues a TWAIT using the same ECBLIST as in a previously completed TWAIT, the problem program should zero out the completion code of any ECB that has been posted complete. This prevents the possibility of the TWAIT being satisfied by a previously posted completion code.

The TWAIT macro instruction is similar to the WAIT macro instruction except that:

- TWAIT requires the completion of only one event before returning control to the problem program.
- The address of the ECB that was posted complete is returned to the user in the register specified (as the first operand of TWAIT).
- The ECB keyword is not used in TWAIT; the ECBLIST keyword is required.
- The on-line terminal test facility is provided if TERMTST is coded in the TWAIT macro (for IBM 1030, 1050, 1060, 2740, the remote 2848, and all binary synchronous devices).

Line Error Print (LERPRT) Macro Instruction

The LERPRT macro instruction provides the ability to print the current values of the accumulators. The accumulators are incremented each time an error threshold value is reached within a specified number of requests for transmission on a line or each time the transmission counter reaches its threshold value without an error counter having reached its threshold value. When the macro instruction is issued, the current values of the error counters and the transmission counter are added to the set of four accumulators and the counters are cleared. After printing takes place, the accumulators are optionally cleared.

Name	Operation	Operands
[symbol]	LERPRT	{dtfbt-symbol (r) (1)} [,rln-absexp ,(r) ,(0)] [,CLEAR=YES ,CLEAR=NO]

dtfbt
: Specifies the symbolic address of the DTFBT, which contains the address of the line error block to be processed.

rln
: Specifies the relative line number to indicate the entry in the line error block to be printed (0-30 allowed, but must be less than "nlines" in the LERB macro instruction). If this operand is omitted, all nonzero entries in the line error block are printed.

CLEAR
: NO specifies that the accumulators will not be reset to zero after the execution of the LERPRT macro instruction.
: YES specifies that the accumulators will be reset to zero after the execution of this macro instruction. If the operand is omitted, YES is assumed.

TRNSLATE (Translate) Macro Instruction

The TRNSLATE macro instruction provides the user with the facility to translate characters from EBCDIC to another transmission code or from the transmission code to EBCDIC. In order to use the TRNSLATE macro, however, the user must have coded TRANSL=YES in the BTMOD macro instruction issued for his program.

Name	Operation	Operands
[symbol]	TRNSLATE	{dtfbt-symbol (r) (1)}, {table-code, (r)} {area-relexp, (r)} {length-absexp (r) (0) 'S'}

dtfbt
    Specifies the address of the DTFBT for the line group.

table
    Specifies the address of the translation table to be used. If a DOS/BTAM table is to be used, the user must code the name of the appropriate table as given in Figure 4 (RSCI, SSCI, etc). Also, the coded table name must have appeared as an operand in an ASMTRTAB macro instruction in the program. The user is permitted to use register notation, in which case he must load the address of the table into the specified register before issuing the TRNSLATE macro. If the address of a user-defined translation table is specified, that table must have been formatted by the user according to the requirements of the System/360 Translate (TR) instruction (See IBM System/360 Principles of Operation, Form A22-6821).

area
    Specifies the address of the first byte of the area containing the characters to be translated.

length
    Specifies the number of bytes to be translated. Any decimal number 1-32767 or equivalent absolute expresssion is allowable. If the "length" operand is coded 'S' (apostrophes mandatory), then the "area" operand is assumed to point to the first buffer of a chain of buffers. All the data characters in the buffers of the chain will be translated. Register notation is permitted, provided the user loads the specified register with the binary value of the length before issuing the TRNSLATE macro. If the specified register contains binary zero, BTAM interprets this to mean the same as a coded 'S'.

ONLTST (On-Line Test) Macro Instruction

The ONLTST macro is used to generate linkage to the on-line test logic in BTMOD. This macro is issued whenever the user wishes to have BTAM initiate binary synchronous on-line terminal test. BTAM initiates on-line test by transmitting an RFT message requesting the remote unit to perform the terminal test that is specified by the parameters of the ONLTST macro. The ONLTST macro can be issued only in communication between S360 and S360, or in any type type of binary synchronous communication when the x operand of the macro is zero.

Name	Operation	Operands
[symbol]	ONLTST	{DECB=symbol (r) (1)}, {X=absexp (r)},{Y=absexp (r)}, [DTFBT=symbol (r)], {AREA=relexp (r)}, [TEXT=relexp, (r) LENGTH=absexp (r)], [ENTRY=relexp (r)], [RLN=absexp (r)]

DECB
    Specifies the address of the DECB associated with the line on which the test is to be performed.

X
    Specifies the type of on-line test to be requested by the on-line test module. This value is used in the X field of the RFT message generated by the test module. Permissible values of X and their meanings are shown below. When specified as ONLTST macro operands, these values of X which are greater than 0, are only applicable in S360-to-S360 communication. For each case where the value of X is greater than zero, the requested message is returned y times.

Y
    Specifies the number of times the requested message is to be transmitted.

DTFBT
Specifies the address of the DTFBT table for the line group.

AREA
Specifies the address of a 300 byte I/O area or an I/O area large enough to accommodate the test message. Buffers may be used, but the first buffer must be large enough to contain the on-line test message or RTF message.

TEXT,LENGTH
The text operand specifies the address of framed text which is to be transmitted during the test. This text will be put in the RFT message only if the x operand is 1. The text must be framed by STX and ETX for normal data. Transparent data must have DLE STX as the first two characters. Ending characters for transparent data are provided by BTAM. The length operand specifies the number of bytes per test and must be coded whenever the text operand is coded.

Note: Transparent data cannot be used in applications where the transparent operations are not defined.

ENTRY
Specifies the address of a DIALST, IDLST, or OPENLST provided by the DFTRMLST macro instruction in the problem program. This operand is coded on switched lines to provide dial digits and/or ID characters to BSC on-line test. For multipoint lines, entry specifies the address of an OPENLST that has the selection characters of the terminal to which the RFT message will be sent. In this case, X=0 is mandatory

RLN
Specifies the relative line number (0-30) within the line group.

The following chart shows the possible values of the x operand and their meanings.

X Value	Meaning
00	Request acknowledgment of the test message Y times
01	Request text received in the RTF transmitted Y times
02	Request transparent EBCDIC message
03	Request USASCII transparency message
04	Request normal EBCDIC message
05	Request normal USASCII message
06	Request alphameric USASCII message
07	Request USASCII printer message
08	Request USASCII punch message
09	Request TRANSCODE printer message
10	Request TRANSCODE punch message
11	Request TRANSCODE multi-point message
12	Request EBCDIC printer message
13	Request EBCDIC punch message
14	Request EBCDIC alphameric message
15	Request EBCDIC weak pattern message for switched line
16	Request EBCDIC weak pattern message for leased line
17	Request TRANSCODE weak pattern message for switched lines
18	Request TRANSCODE weak pattern message for leased lines
19	Request EBCDIC weak pattern for DLE SYN insertion

BUFFER MANAGEMENT MACROS

The macro instructions, REQBUF and RELBUF, provide the execution time facility to obtain buffers from and release buffers to the BTAM-maintained buffer pools.

REQBUF (Request Buffer) Macro Instruction

The REQBUF macro instruction permits the user to obtain buffers from the buffer pool of the line group. (See the section, Buffering, for more details on buffers.)

Name	Operation	Operands
[symbol]	REQBUF	⎧dtfbt-symbol⎫ ⎨(r)⎬,(r) ⎩(1)⎭ ⎡,count-absexp⎤ ⎢,(r)⎥ ⎣,(0)⎦

dtfbt
Specifies the address of the DTFBT table in which the buffer pool is defined, via the DTFBT operand BUFCB.

(r)
Specifies the register in which the address of the first buffer in the chain is to be returned by BTAM to the problem program.

count
Specifies the number (1-255) of buffers requested. If the count operand is omitted, one buffer is assigned.

Cautions: A buffer pool must have been previously assigned to the DTFBT and the DTFBT must be open. Buffers should be returned to the pool by the RELBUF macro instruction when the user has finished using them.

RELBUF (Release Buffer) Macro Instruction

The RELBUF macro instruction permits the user to return to the buffer pool of the line group one or more buffers which are no longer required.

Name	Operation	Operands
[symbol]	RELBUF	⎧dtfbt-symbol⎫ ⎨(r)⎬,(r) ⎩(1)⎭

dtfbt
Specifies the address of the DTFBT table in which the buffer pool is defined, via the DTFBT operand BUFCB.

(r)
Specifies the register containing the address of the first buffer in the chain of buffers to be returned to the buffer pool. All buffers to be returned to the buffer pool must be forward chained. If the user has altered the normal BTAM chaining, he must place in bytes 9 through 11 of each buffer the address of the next one in the chain. The last buffer in the chain to be returned must contain zeros in bytes 9 through 11.

This section provides a general discussion of the steps to be taken in writing a problem program using BTAM.

ASSEMBLY CONSIDERATIONS

Use of the SEPASMB Operand

In a standard Tele-processing application using BTAM, it may be desirable to assemble separately and to catalog individually the various logical components comprising the total programmed support for the system configuration. This technique has the advantage that if one component (say, the problem program) is to be reassembled and cataloged, the others (e.g., the DTFBT's and the BTAM module) need not be. For this reason, both the DTFBT and the BTMOD macro instructions include the SEPASMB keyword operand to give the user this facility.

To make effective use of the SEPASMB operand, the user should be aware of the following considerations:

1. If a BTAM macro is coded with SEPASMB=YES, a CATALR librarian control statement is generated as the first output record of the macro expansion. For this reason such a macro call should appear normally as the first source statement to be assembled. No other macro call with SEPASMB=YES coded should appear in the same assembly.

   Coding of SEPASMB=YES in the BTAM macro also results in the creation of a control section in the macro expansion. The macro expansion does not restore the user's control section following the generated code because it is presumed that a user's control section has not been initiated. Other source code included in the same assembly will be included in the same control section initiated by the BTAM macro unless a named CSECT statement follows the BTAM macro call.

2. If SEPASMB=NO is coded in a BTAM macro call, no CATALR statement is generated. Such a macro call need not appear as the first source statement in an assembly. Any number of macro calls with SEPASMB=NO may be included in a single assembly.

   Although the macro expansion also creates a control section for SEPASMB=NO, the control section preceding the macro call is restored following the macro expansion.

In abnormal usage (e.g. when assembling several DTFBT macro instructions together in order to share a buffer pool and cataloging them as a single relocatable library book) the operand must be coded SEPASMB=NO or omitted in each of the macro instructions. The user must prepare his own CATALR statement to go with the object deck from the single assembly. The catalog name must be the same as the name of some entry point or CSECT in the assembly, which in turn must be the same as some name in an EXTRN statement or V-type address constant in the application program with which the catalogued relocatable module is to be used. This allows the linkage editor AUTOLINK facility to locate the DTFBT book in the relocatable library.

Principles of BTAM Register Usage

Because DOS/360 does not permit the user to make free use of registers 0, 1, 14, and 15, BTAM does not preserve any user's values of these registers except the value placed in register 14 by a linkage to BTMOD. BTAM always preserves the user's values of registers 2-13,* inclusive. However, BTMOD requires the use of registers 1-15 as base registers or DSECT registers. Consequently, BTMOD employs USING and DROP statements involving these registers. This can cause loss of user's program addressability if BTMOD is introduced into the middle of the user's assembly. This condition can be avoided if the user does one of the following:

1. Assembles BTMOD separately, and catalogs it in the relocatable library, or

2. Assembles BTMOD into his program before his first USING statement, or

----

*Register 13 is preserved because compiler-generated DOS routines require register 13 as a save area address register. A save area register is not a universal DOS . requirement, therefore BTAM does not use it for this function. Registers 2-12 are allowed as operands in BTAM macro instructions where register notation is permitted.

3. Assembles BTMOD into his program after his other assembler statements (which should conclude with LTORG), and just before his END statement.

DEFINE THE LINE GROUPS

A line group is defined through a table generated by the DTFBT declarative macro instruction. This table contains information concerning the line connection, the control unit attached to the channel, the remote device type on the lines, and various hardware and software features used for the line group. Other information in the table concerns various programming options and address constants required by BTAM in order to perform the telecommunications functions desired by the user.

The individual lines in the line group are defined by the LINELST keyword operand in the DTFBT macro instruction generating the table. In coding this operand, the user associates each line in the group with a logical (or symbolic) unit which, in turn, is associated with a physical unit via a logical unit block (LUB) in a device table. Each LUB (a 2-byte entry in the device table) contains a pointer to a physical unit block (PUB) which contains information such as channel, unit, device type, etc.

The association of an LUB with a PUB is established for an installation either at system generation time or at problem program execution time. In the former case (system generation time), certain conventions are established to be followed by all programmers. In the latter case, the problem programmer (background programs only) can include ASSGN control statements with his JOB control cards to assign a specific logical unit (SYSnnn) to a specific physical unit. The ASSGN control statement may also be used to override the standard assignments defined at system generation time. The operator (for background or foreground programs) also may make assignments. A full discussion of device assignment may be found in IBM System/360, Disk Operating System, System Control and System Service Programs, Form C24-5036.

DEFINE THE TERMINAL LISTS

The DFTRMLST macro instruction defines terminal lists to be used in polling and addressing terminals.

A single terminal list is normally defined for each line for all read operations. This list includes the polling or invitation-to-send codes for all terminals on the line. The order in which sending devices on a line are contacted to send data is determined by a polling list containing one or more entries for each device to be contacted. A sending device may appear one or more times in a polling list so that the stations requiring more service will be polled more frequently.

Normally, a separate list is defined for each terminal for all write operations. Each of these lists includes the addressing or call-directing code for the terminal that is to receive the message. All receiving devices must be defined in addressing lists. An addressing list may include addresses for more than one terminal or component when multi-addressing is used. BTAM contacts all terminals in the list before sending the message. Each terminal sends back a response which indicates whether it is ready to receive or not. If any terminal fails to answer or sends a negative (not ready to receive) response, the message is not sent. An addressing list is always of the open type.

When terminals on different lines have the same polling and addressing characters, the same polling and addressing lists may be used for these lines. A single list may be used by more than one line at the same time if the operation is the same for the lines. Obviously, if CHGNTRY is issued for a terminal list, polling and addressing on all lines using the list would be affected.

The terminal devices connected to an audio response unit are not defined in terminal lists. Line connection is established on the subscriber's request, and no polling or addressing takes place. Once the line connection is established, exchange of information takes place.

The sections on the individual terminal types in this publication illustrate the particular DFTRMLST formats used for reading and writing on a given terminal/network configuration. Appendix A gives examples and descriptions of the generated lists.

The only choice left to the programmer is the type of list that he wants to use for polling terminals on non-switched networks. For these terminals, he can choose either an open list, a wrap-around list, or an Auto Poll list.

If an open list is provided for the READ macro instruction, polling ceases when the last entry in the list is polled. Each component is polled in turn, beginning with the entry whose address is given in the READ macro instruction. For each terminal or component polled, the terminal control unit returns a response. A negative response indicates that the terminal or component has no message to send to the computer. A positive response consists of the message itself. The first positive response returned stops polling. Polling of successive components ends when either:

1. a positive response is returned;

2. an error condition from which BTAM cannot recover is detected;

3. a RESETPL macro instruction is issued; or

4. the end of the list is reached.

When transmission is completed, or one of the last three above conditions exists, the read operation is posted complete.

If a wrap-around list is provided, polling continues indefinitely until either:

1. a message is received from the terminal;

2. an error condition from which BTAM cannot recover is detected; or

3. polling is terminated by the RESETPL macro instruction (see the RESETPL macro instruction description).

When polling on a wrap-around list is terminated by either condition 2 or 3, or when data transmission is complete, the read operation is posted complete. When a wrap-around list is used, the READ Initial macro instruction may specify an 'S' as the terminal list address. In this case, BTAM will poll either:

1. the same terminal, if the last polling sequence produced a message or an error condition; or

2. the next active terminal in the list, if the last polling sequence produced a negative response and was stopped by a RESETPL macro instruction.

This allows the problem programmer to design a system that obtains all messages to be sent from a polled terminal, and permits ease of error retry.

DEFINE THE DECB'S

A data event control block is normally defined for each line. This may be done using the MF=L or the MF operand omitted form of the READ, WRITE, or CONTROL macro instruction. Normally, a single DECB per line is sufficient, but additional ones for specific purposes may be desired. The DECB contains data that must be used both by the programmer and by BTAM. See Figure 15 for the format of the DECB.

BTAM posts in the DECB the event associated with that DECB. An event which satisfies a WAIT condition may be one of the following:

1. A successful completion of a READ, WRITE, CONTROL, or RESETPL operation.

2. A program-controlled interruption (7772 only).

3. A nonsuccessful operation, as follows:

   a. Polling - when the end of an open list is reached, the polling stops.

   b. Addressing - when the addressed device is not ready, no message is sent.

   c. Multi-addressing - when one of the addressed devices is not ready, the message is not sent to the ready devices.

   d. An error condition.

4. A completion of a RESETPL macro instruction.

	0	1	2	3
0	Completion Code	Reserved		
4	Optype Qualifier Bits	Optype Code	Length	
8	Response Information Byte	DTFBT Address		
12	Mode Byte	Input/Output Address		
16	Sense Byte	Sense Byte for Diagnostic Write/Read	Residual Count	
20	Command Code	List Address or Entry Address		
24	Flag Byte	Relative Line Number	Response to Addressing	Response to LRC
28	TP Code	Error Information	Status Bytes	
32	Reserved	Addressing Pointer		
36	Index Byte	Polling Pointer		

Figure 15. Data Event Control Block

The following describes the DECB fields in Figure 15 for the DECB.

Bytes 0-3: Event Control Block (ECB). BTAM sets byte 0 to zero prior to execution of a macro instruction and gives a completion code in that byte when an event has been completed. (See the section on completion codes.)

Byte 4: Optype qualifier bits. This byte contains various flags which are set ON as follows:

bit-0 Initial optype coded

bit-1 "R" added to optype of READ/WRITE when Reset option is used (not applicable for WRITE TR)

bit-2 "L" added to optype of READ/WRITE for 2260 Local when Lock option is used.

bit-3 Conversational WRITE coded (BSC only)

bit-4 Start-stop Auto Poll used in current READ

bit-5 Entry operand coded as 'S'

bit-6 Area operand coded as 'S'

bit-7 Length operand coded as 'S'

Byte 5: Optype code. This byte contains a hexadecimal value corresponding to the optype coded in the READ, WRITE, or CONTROL as follows:

Hex Value	Optype
00	WRITE TB
00	WRITE TD
01	READ TI
02	WRITE TI
03	READ TT
04	WRITE TT
05	READ TV
05	READ TTL
06	WRITE TV
07	READ TP
08	WRITE TA
08	WRITE TIE
09	READ TIQ
09	READ TS
0A	WRITE TN
0A	WRITE TIX
0B	READ TB
0B	READ TPL
0C	WRITE TL
0C	WRITE TIXV
0D	READ TO
0D	READ TQ
0E	WRITE TS
0E	WRITE TIV
0F	READ TTS
10	WRITE TC
10	WRITE TCO
10	WRITE TE
11	READ TRV
12	WRITE TX
12	WRITE TIO
14	WRITE TXV
16	WRITE TR
18	WRITE TW

1A	WRITE TQ
1B	CONTROL TI
1C	CONTROL TD
1D	CONTROL TM

Bytes 6-7: Length. This is the length specified by a READ or WRITE macro instruction length operand.

Byte 8: Response Information Byte.
Bits 0-6, reserved.
Bit 7 is ON when the SAK character is received by the transmitting CPU as response to text.

Bytes 9-10-11: DTFBT address. This is the address of the DTFBT table associated with the DECB.

Byte 12: Mode byte: This byte is used to store information for the Set Mode command for binary synchronous communication.

Bytes 13-14-15: Input/output area address.

Byte 16: Sense byte. This byte contains the sense byte after a unit check condition occurs.

The bits are set ON when the following conditions occur:

Bit 0	Command reject
Bit 1	Intervention required
Bit 2	Bus out check
Bit 3	Equipment check
Bit 4	Data check
Bit 5	Overrun
Bit 6	Lost data
Bit 7	Time out

Byte 17: Sense byte for a 2701 diagnostic Write/Read.

Bytes 18-19: Contains the residual count for the read or write data command; otherwise, it will be zero.

Byte 20: Command code. This identifies the type of command upon which the error occurred.

Bytes 21-22-23: Entry address.

Byte 24: Flag byte. The bits are:

Bit 0 is ON when PCI occurs (7772 only)
Bit 1 is ON if EOT is received (binary synchronous communication only)
Bit 2 is ON if DLE,EOT is received (binary synchronous communication only)
Bits 3-5, reserved
Bit 6 is ON if the ECB contains the address of the first unreleased buffer not used at the completion of a READ (whose area operand is specified as 'S')
Bit 7 is ON when a terminal test is operating on the line.

Byte 25: Relative line number. This is the relative number of the line currently associated with the DECB.

Bytes 26-27: Response to addressing field. For start-stop transmission, the response to addressing character(s), if any, is read into byte 26. The response to the longitudinal redundancy check (LRC), if any, is read into byte 27. For binary synchronous communication, both bytes are used for responses.

Byte 28: Contains the TP code of the last executed command in a channel program that has terminated. These TP codes may be used by the problem program to identify the function of a command in a channel program when the function is not identifiable by the command code alone.

The hexadecimal TP codes have the following meaning for start-stop channel programs:

00 Null

01 Disable (if first command of channel program) Dial, Enable, Write pad characters, Prepare.

02 Write Ⓓ Ⓒ Ⓒ Ⓒ (prior to selection), Write EOT sequence (prior to polling or addressing).

03 Write polling, or addressing characters, Write turn-around sequence (TWX), Write CPU-ID sequence (TWX), or Poll.

04 Write space, Write code (2260R), Write shift (83B3), Write one (1030).

05 Read response to polling.

06 Read response to addressing.

07 Read ID response (TWX).

08 Write end-of-addressing character following addressing.

09 Write response to text, Write Ⓓ and 15 idle characters.

0A Write idle characters.

10 Write at Line Address (2260).

11 Read or Write text.

12 Read or Write in buffers.

20 Read response to text.

21 All Reset commands (i.e., those commands in a channel program that perform the reset function).

22 Read Skip.

23 Write Break.

The TP codes used in BSC channel programs are:

00 Not used.

01 Prepare, Enable, Dial.

02 Write special characters (DLE ETB, DLE ETX, WABT), Poll (at beginning of channel program).

03 Poll, Write addressing characters, Write ID, Write ENQ, TIC (to a Poll), I/O No-Op (in an Auto Poll channel program).

05 Read response to ENQ.

06 Read response to addressing.

07 Read ID, ACK-0.

08 Read ID,ENQ.

09 Write response to ENQ, Write response to text.

11 Read text, Write text, TIC (to a buffer).

12 Read text into buffers, Write text from buffers.

14 Read response to SAK (Stop Acknowledgment).

20 Read response to text.

21 Write EOT, Write DLE EOT, Set Mode, Disable.

24 Read ENQ.

31 Read Skip (ERP).

32 Write ENQ (ERP).

34 Read ENQ (ERP).

35 Write DLE ENQ (ERP).

36 Disable (ERP).

39 Write ACK-0/ACK-1 (ERP), Write NAK (ERP).

Byte 29: Error information field. This field provides the following information if ERP is specified:

Bit 0, reserved
Bit 1, should not occur error
Bit 2, error in ERP
Bit 3, diagnostic write/read failed
Bits 4-5, not used
Bit 6, a message of the form: STX--Text (may be zero length)--ENQ received from remote device (2780 or Model 20). Bit 1 of the Flag byte will also be ON when this occurs.

Bit 7, NAK received in reply to transmitted message or as the reply to WRITE Inquiry (binary synchronous communication only).

Bytes 30-31: Status bytes. These bytes contain the status bytes after a completion occurs.

Byte 32: Reserved.

Bytes 33-34-35: Addressing pointer.

Byte 36: Index byte. At the completion of a READ Initial using start-stop Auto Poll, this byte contains the index byte of the last entry in the list to be polled.

Bytes 37-38-39: Polling pointer. At the completion of a READ for the 2260 Local, the rln for the display station from which data has been read is placed in the low-order byte of the polling pointer field (byte 39).

BUFFERING

The buffer management logic consists of routines to:

1. Perform the functions requested through the REQBUF and RELBUF macro instructions (for obtaining and releasing buffers).

2. Read incoming messages into buffers (when the user codes the area operand as 'S' in a READ macro) or write outgoing messages from buffers (when the user codes the length operand as 'S' in a WRITE macro).

The buffer management logic, however, is optionally included in the BTAM module for a problem program. For example, if BUFFER=NO is specified in the BTMOD macro for the problem program, no buffer management is possible for that program. If BUFFER=REQREL is specified, the routines to perform (1) above are included for the program, but the routines to perform (2) are not. On the other hand, if BUFFER=YES is coded, the routines to accomplish both (1) and (2) are included for the problem program.

Through the DTFBT macro, the problem program may, if the buffer management routines are included for the program, define for the line group whether or not buffering is to be done. This is done by coding the BUFCB, BUFNO, and BUFL operands in the DTFBT macro. The BUFNO and BUFL keyword operands in DTFBT allow the user to define the number of buffers and the length of each buffer in a buffer pool. The user may also share a buffer pool between two or more line groups.

For a line group for which buffering is specified, the BTAM OPEN routine will organize the buffer pool automatically (provided it is not a shared buffer pool previously organized and/or currently being used by another line group).

Buffer Pool

The first eight bytes of the buffer pool are the buffer control block (see Figure 16).

c - Buffer pool counter; indicates the number of currently open line groups using the buffer pool.

a - Address of the first buffer available.

nd - Number of buffers as defined in the DTFBT macro instruction via the keyword operand BUFNO.

na - Number of buffers currently available. Initially, na = nd.

length - Length of each buffer as defined in the DTFBT macro instruction via the keyword operand BUFL.

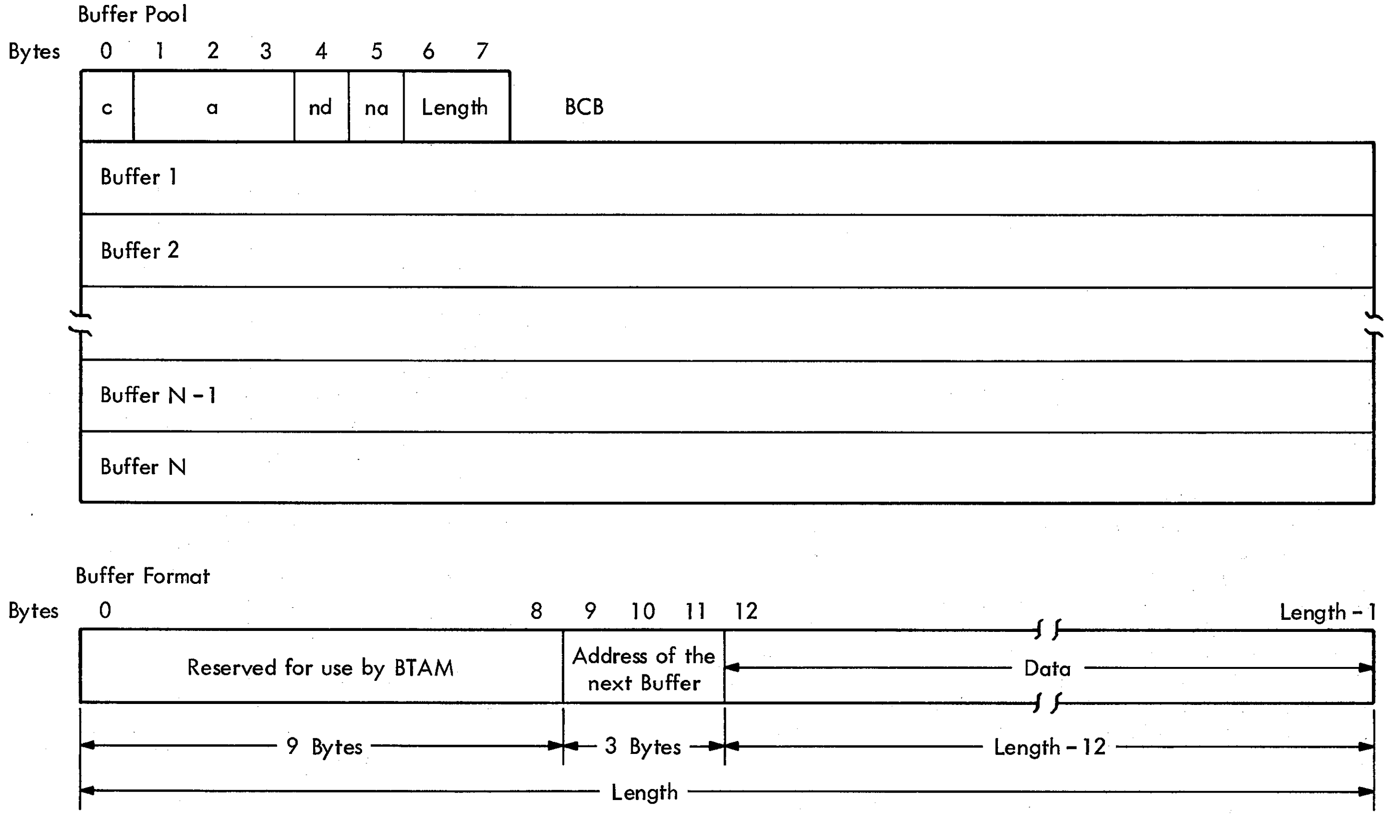


Figure 16. Buffer Pool and Buffer Formats

Each buffer pool and each buffer within the pool begins on a doubleword boundary. The problem program makes better use of main storage if it requests buffers whose lengths are multiples of eight bytes.

Buffer Format

Each buffer has the format shown in Figure 16.

Bytes 0 through 8 are reserved for use by BTAM for channel commands.

Bytes 9, 10, and 11 contain the address of the next buffer in the chain. If this buffer is the last buffer in the chain and the Reset function is not specified, bytes 9, 10, and 11 contain zero. If this buffer is the last buffer in the chain and the Reset function is specified, the address of the Reset CCW is contained in bytes 9, 10, and 11.

Bytes 12 through length-1 contain:

1. Zero when the buffers are in the pool and when the user gets them by a REQBUF macro instruction.

2. The data read, if a READ (area operand 'S') has been issued.

3. The data to be written, when a WRITE (length operand 'S') is issued.

Buffers for Read Operations

If the area operand of the READ macro instruction is coded 'S', BTAM will provide the user buffers from the buffer pool of the line group. The length operand of the READ macro instruction specifies the maximum length in bytes of the expected message. BTAM will provide as many buffers as necessary, provided the required number is in the pool. These buffers will be chained. Once the operation is complete, the address of the first buffer in the chain is in the area field in the DECB for the line (bytes 12 through 15).

For the binary synchronous communication Conversational-type WRITE's (see WRITE Initial Conversational in the section CPU-to-CPU Contention), BTAM also provides buffers (optionally) for the received message. In this case, the address of the first buffer in the chain is placed in Word 1 of the user's parameter list rather than in the area field of the DECB (which contains the address of the user's output area).

The user must return the buffers whose messages have been processed. The release buffer (RELBUF) macro instruction releases buffers to the buffer pool.

If an error is posted on a Read operation using buffers, the user must return the acquired buffers to the pool before retrying the operation.

Buffers for Write Operations

A request buffer (REQBUF) macro instruction obtains a chain of buffers from the buffer pool. The address of the first buffer in the chain is returned in the register specified by the second operand in the REQBUF macro instruction. If the length operand in the WRITE macro instruction is coded 'S', BTAM will write from a chain of buffers. The user must specify the address of the first buffer in the area operand of the WRITE macro. To avoid having BTAM write out the full contents of the last buffer in the chain (when no end character is in the buffer or the end character does not terminate the Write operation), the user can use a special technique. The exact count of data characters in the last buffer can be placed in bytes 6-7 (the 7th and 8th physical bytes) of the last buffer in the chain prior to issuing the WRITE macro.

At the completion of the WRITE macro instruction, the user must return the buffers which are no longer required. The release buffer (RELBUF) macro instruction is used to release the buffers.

Buffer Pool Updating

Step 1 - At assembly time, the DTFBT macro instruction creates the buffer control block and reserves main storage for the buffer pool.

Step 2 - At OPEN time the pool is organized, with all buffers cleared and chained forward. Assume for discussion that first buffer available is a which is chained to b chained to c chained to d, etc.

Step 3 - If the buffers a and b are taken from the pool by a READ operation or a REQBUF macro instruction, the next buffer available is c which is chained to d, etc.

Step 4 - If the buffers m, n, o, and p are released to the pool by a RELBUF macro instruction, and the first buffer available is j, then buffer p is chained to j and buffer m becomes the first buffer available.

Figure 17 illustrates steps 2, 3, and 4.

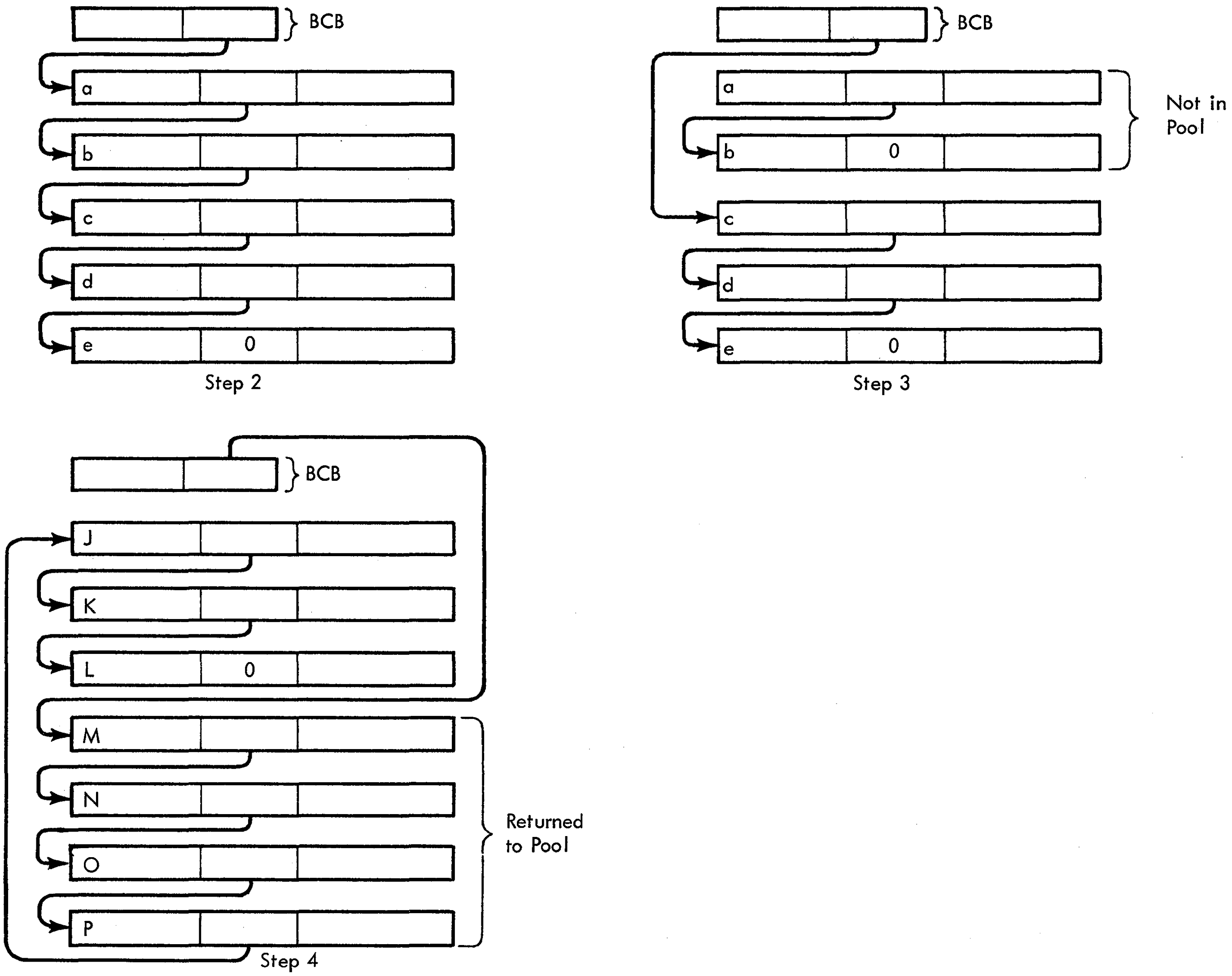


Figure 17. Buffer Pool Updating

The user requests buffers by issuing a REQBUF macro instruction. If the number of buffers available is exceeded by the number requested, the available buffers are still passed to the user. A return code indicating that there were either not enough buffers or not any buffers is returned in the low-order byte of register 15. The number of buffers exceeding the number available is in register 0. See the section on return codes for REQBUF.

The user may also request buffers implicitly by issuing a READ macro instruction (area coded as 'S'). If the number of buffers available is not sufficient to contain the maximum input message, then the request is ignored. In register 15 there is a return code of hexadecimal 18 indicating too few buffers available.

POSTED ERROR INFORMATION

BTAM provides information to help the user in recovering from error conditions occurring during the execution of a BTAM operation. These conditions may be caused by problem program errors or by temporary or permanent failure of the data link.

The error information provided by BTAM includes return codes for problem program errors and completion codes on I/O termination. The conditions of the error are posted in the DECB. For the audio response units, certain error recovery procedures are provided as standard facilities. If the ERP option has been chosen, the operation in error is retried if possible and system-to-operator messages are sent when the number of retries reaches a retry-limit or when the error cannot be corrected. In the latter case (with ERP), comprehensive error information is posted in the DECB.

Return Codes: When BTAM detects a problem program error in the issuing of a READ, WRITE, CONTROL, REQBUF or RELBUF macro instruction, the macro is not executed, the error condition is recorded, and control is returned to the problem program at the instruction following the macro in error.

BTAM records such errors by setting the condition code to a nonzero value (so that the problem program may use a branch-on-condition instruction) and by setting a return code in bits 24 through 31 of the general register 15 (bits 0 through 23 being zero). Note that if no error has occurred, register 15 contains a return code of X'00'.

The return codes set for READ, WRITE, and CONTROL are shown below.

Code in Hex — Meaning

04 Busy. The specified line (or DTFBT for 2260 Local) is busy with a previously requested operation.

08 Invalid relative line number. There is no line within the line group with a relative line number equal to that specified by the user in the macro instruction.

0C Invalid type code. The code specified in the optype operand is invalid for the device.

10 All skip bits on. The skip bit is ON in all of the entries in the polling or addressing list specified in the macro. This return code is also used if there are no active entries in the Auto Poll list.

14 Line error at OPEN. The user should cease reference to that line until he can correct any physical difficulties with the line or data set. He may then issue an LOPEN macro instruction to retry to open the one line.

18 No buffers. There are not enough buffers available in the buffer pool to satisfy the needs of a READ macro instruction with the area operand coded 'S' or a Conversational-type WRITE (binary synchronous communication) for which buffers are requested.

1C No buffer pool. The user issued a READ macro instruction with the area operand coded 'S' or a WRITE macro with length coded 'S' or a Conversational-type WRITE having Word 1 of the specified parameter list containing a zero address but the DTFBT specified does not contain the address of a buffer pool.

20 No buffer management. The user issued a READ macro with area coded 'S' or a WRITE macro with length coded 'S' or a Conversational-type WRITE having Word 1 of the specified parameter list containing a zero address but the BTMOD macro for the problem program did not specify BUFFER=YES.

Note: All non-zero return codes indicate that the operation was not properly performed. The problem program must not issue WAIT macro instructions for READ, WRITE, or CONTROL macros that result in nonzero return codes, because no I/O operation is initiated.

The return codes set for REQBUF and RELBUF are:

Code in Hex — Meaning

04 Partially performed. Some, but not all, of the buffers have been provided to the user, or returned to the buffer pool. Register 0 contains the count of buffers not supplied or not returned to the pool. For RELBUF, the user-supplied register contains the address of the first buffer not returned.

08 Not available. None of the buffers have been provided to the user, or returned to the buffer pool. For REQBUF, the user-specified register will contain zero, and for RELBUF the user-specified register will be unchanged while register 0 will contain the count of the buffers not supplied or not returned.

0C No pool. The DTFBT specified does not contain the address of a buffer pool.

10 No buffer management. The BTMOD macro for the problem program did not specify either BUFFER=REQREL, or BUFFER=YES.

Note: If the DTFBT specified in the REQBUF/RELBUF is not open, a system-to-operator message is written and the job is canceled if the user's BTMOD specified CANCEL=YES. If CANCEL=NO, control is returned with a cancel return code of X'01'.

The return codes set for CHGNTRY (with SSALST and SSAWLST only) are:

Code in Hex	Meaning
00	Action was performed or was not needed. For example, if SKIP was specified and the entry was not found to be active, then the action was not needed.
04	Action was not performed because CHGNTRY was issued for a terminal list which is being polled.
08	Action was not performed because the position operand is too large. The value of the position must be less than the number of entries.

The nonzero return code set for TWAIT is:

Code in Hex	Meaning
04	TERMTST was coded in TWAIT but terminal test logic was not included in BTAM logic module.

The nonzero return codes set for TRNSLATE are:

Code in Hex	Meaning
01	The length operand indicated buffers but the associated DTFBT was not open. This return code is given only if CANCEL=NO appeared in BTMOD. For CANCEL=YES a system-to-operator message is given and the job is canceled.
04	TRANSL=NO was coded in BTMOD; no translation can be performed.
08	The length operand indicated buffers but no buffer pool is associated with the DTFBT.

The return codes set for ONLTST are:

Code in Hex	Meaning
00	Normal return with no error.
04	The specified line is busy with a previously specified operation
08	Relative line number is too large
0C	Invlaid text code. The value specified by the X operand of the ONLTST macro is undefined.
10	Illegal use of ONLTST macro.
14	Line error at OPEN time.

Completion Codes: Upon termination of a read, write, or control operation, BTAM sets a completion code in bits 0 to 7 of the ECB that indicates the nature of the termination. The completion codes follow.

Code in Hex	Meaning
7F	Normal completion. BTAM has not recognized any I/O failure due to the hardware or the problem program.
41	I/O error. The following information will be posted in the DECB: Bytes 16-17: For the 2701 byte 16 is the sense byte and byte 17 is the sense byte (if ERP is present) for diagnostic Write/Read. For the other control units the sense byte is posted in byte 16. Bytes 18-19: Residual count of the channel command in error. Byte 20: Command code of the channel command in error. Byte 28: TP Code from the channel command in error. Byte 29: Error Information Field (posted only when ERP is present, except for Bit 1 which is posted as an "unknown error" bit when ERP is absent). Bytes 30-31: Status Bytes Note: For binary synchronous communication, byte 24 (Flag byte) is also posted when EOT or DLE,EOT is received. This may occur in conjunction with irrecoverable errors or normal completions (7F).
42	READ macro completed with no text transfer (2260 Local only). This completion code is posted if the 2260 Local operator neglects to press the START key before keying in text and pressing the ENTER key or if the START key is depressed immediately before the ENTER key.

Note: Completion code 42 is also posted if an RFT message is received on a Read Initial operation but no TWAIT macro was issued by the BSC proproblem program. In this case the problem program should now issue a TWAIT macro on the ECB without removing the X'42' completion code from the ECB.

44 Received ID did not compare. Applies to WRITE TI for AT&T Model 33/35 TWX Terminals and to binary synchronous CPU-to-CPU Dial operation. It indicates that the ID received on a dial operation did not compare with the expected ID defined in the terminal list. The condition could result from:

1. specification of an incorrect ID or telephone number in the terminal list, or
2. a transmission error which resulted in one or more of the ID characters being garbled, or
3. a transmission error or switching exchange error which resulted in dialing an incorrect number (the wrong terminal).

Note: Completion code 44 indicates, for BSC, that initial contact was not successfully established between stations on leased, dial, or multipoint lines due to a garbled "bid" (ENQ) or an invalid or negative response to a bid for the line.

48 Channel program halted as a result of a RESETPL macro instruction being issued.

50 Contention. indicates that the slave CPU lost the line to the master CPU while attempting to transmit.

54 Nonproductive operation that indicates:

1. a negative response to polling has been received from the terminal represented by the last active (non-skip) entry in an open polling list (OPENLST),
2. a negative response to polling has been received following a RESETPL instruction,
3. all of the entries in a wrap-around polling list (WRAPLST) are inactive (all skip bits are ON), or
4. a negative response to polling has been received from the last entry in an Auto Poll list. This completion code is used for the Auto Poll of an AUTOLST type list and for the Auto Poll of the AUTOWLST type when a RESETPL has been issued.
5. a negative response to addressing (start-stop only) has been received.

Note: Condition 3 can occur only as a result of one or more skip bits being turned ON after initiation of a polling operation (READ TI for all terminals, except TWX terminals, with a wrap-around polling list). If all skip bits were ON at the time that the READ macro instruction was executed, no I/O operation would be initiated.

58 Cancel condition detected.

60 Alternating Acknowledgments. indicates that an incorrect positive response was received. (ACK-0 instead of ACK-1 or vice versa).

61 WABT Received. indicates the WAIT-Before-Transmitting characters were received as a reply to a message or in response to ENQ.

62 Enquiry Character (ENQ) received in place of a message. This means the last acknowledgment sequence should be retransmitted to the remote CPU. If the ENQ character is received in response to a text message sent, this completion code will also be posted. This is an abnormal line control procedure which can occur in binary synchronous communication when "conversational" data exchange (see WRITE Initial Conversational in CPU-to-CPU Contention) occurs. When an ENQ is received in response to an ENQ after initial connection has already been successfully established, this completion code is also used to inform the user of the abnormal condition.

When the ENQ is expected in CPU-to-CPU (Contention and Dial systems) communication for the purpose of establishing initial connection, this completion code is not used.

SERVICEABILITY FACILITIES

BTAM provides the user with serviceability facilities on an optional basis. The user specifies which facilities will be included in the BTAM logic module by coding the appropriate operands in the BTMOD and DTFBT macro instructions.

ERROR RECOVERY PROCEDURES

Error recovery procedures (ERP) are provided on a line group basis. ERP is a standard facility (default options ERLOGIC=E in BTMOD and ERROPT=E in DTFBT), however, the user may omit ERP by coding ERLOGIC=N in BTMOD and ERROPT=N in DTFBT and write his own ERP, making use of the information posted in the DECB, except for binary synchronous communication and audio response where ERP is mandatory.

If the operand ERLOGIC=E is coded in BTMOD or omitted, the user may specify the type of error recovery he wants by coding in the DTFBT:

1. ERROPT=E for error recovery alone
2. ERROPT=W for ERP with write-text retry
3. ERROPT=R for ERP with read-text retry
4. ERROPT=RW for ERP with full text retry

BTAM will retry transmission when feasible.

When an interruption occurs in a channel program initiated by BTAM, the supervisor gives control to the BTAM Interruption Handler. Note that in the event of a channel control check or an interface control check, the supervisor cancels the job (See IBM System/360 Disk Operating System, Supervisor and Input/Output Macros, Form C24-5037.)

BTAM Interruption Handler checks the conditions of the interruption. If the interruption is a normal condition, BTAM Interruption Handler takes the proper action (i.e., post completion, restart polling, etc.).

If the interruption is an abnormal condition and ERP is specified, BTAM tries to recover from the error. If the attempt to recover is successful, normal completion is posted in the DECB. If the attempt to recover is unsuccessful, a system-to-operator message is typed. The format of the message is as follows:

```
          {C}
[pp] 4BnnI{P}Text SYSyyy=mmm DECB=aaaaaa
TI=xxxx DC=hhhhhhhhhh
[pp] 4B99I CSW=**eeeeeeeeeeeeee
CCW=dddddddddddddddd SN=ffff
```

- pp for partition indicator, supplied by DOS only with a multiprogramming supervisor
- 4B specifies BTAM message
- nn for the decimal error code (Figure 18)
- I for operator's action (information)
- C indicates action (cancel)
- P indicates action (post)
- text for the explanation of the error
- yyy specifies the symbolic unit
- mmm specifies the line associated with the symbolic unit
- aaaaaa for the DECB address in hexadecimal
- xxxx for the terminal identification in hexadecimal. This field is meaningless and should be disregarded for the following devices: Audio Response Units, TWX terminals, 2740 without station control, and all point to point BSC devices.
- hhhhhhhhhh for the dial characters
- eeeeeeeeeeeeee for the last seven bytes of the CSW in hexadecimal
- ddd...d for the CCW in error

NOTE: the CCW contents are as follows:

Bytes(s)	Contents
0	Command code
1-3	Data address
4	Flags
5	TP code
6-7	Byte count

- ffff for the contents of the Sense byte

Figure 18 gives the code and explanation of the ERP messages.

If the operand CANCEL=YES is coded in BTMOD (see BTAM Cancellation), the job is canceled for:

- Channel Data check
- Program check
- Protection check
- Command reject

The conditions of the error are posted in the DECB.

Error Code	Text	Explanation
20	ERR IN ERP	Error in ERP
21	CHAN DATCK	Channel data check
22	SHOULD NOT	Should not occur error
23	CHAIN CHK	Chaining check
24	PROGRAM CK	Program check
25	PROTECT CK	Protection check
26	UNIT EXCEP	Unit exception
27	EQUIPMT CK	Equipment check
28	LOST DATA	Lost data
29	TIME OUT	Time out
30	INTERV REQ	Intervention required
31	BUS OUT CK	Bus out check
32	DATA CHECK	Data check
33	OVERRUN	Overrun
34	COMMAND RJ	Command Reject
35	STX ENQ	Error in actions of problem program (if terminal is not 2780 or Model 20) Hardware malfunction on 2780 or Model 20 (See Operating Guide or 2780 Component Description)
36	STX T ENQ	Error in actions of problem program (if terminal is not 2780) Line buffer parity check or line buffer overrun on 2780
37	EOT RESPN	Error in actions of problem program (if terminal is not 2780 or Model 20) Hardware malfunction on 2780 or Model 20 (See Operating Guide or 2780 Component Description)
99	-- --	Appears with each of the above message codes to give more information

Figure 18. ERP Error Messages

ERROR RECOVERY SUGGESTIONS

Use of the Error Recovery Procedures is strongly recommended. The operation in error is retried if possible, and the error conditions are posted in the DECB. System-to-operator messages inform the operator when an error cannot be recovered or when the number of retries per terminal exceed a user-provided threshold.

This section provides suggestions for the DOS/BTAM user who intends to write his own error recovery routines. These suggestions must be followed carefully. However, the user must be aware that the lack of automatic retries for the operations in error would have a large incidence on the throughput of the system. Lack of information provided to the operator inhibits proper maintenance of the system.

The Standard Error Information posted in the DECB provides enough information to identify the error.

If the interruption is an abnormal condition, the BTAM Interruption Handler takes the following actions:

- Stores the status bytes in bytes 30 and 31 of the DECB.
- Stores the command code of the channel command in error in byte 20 of the DECB.
- Stores the TP code of the channel command in error in byte 28 of the DECB.
- Stores the residual count in bytes 18-19 of the DECB.
- Stores the sense byte in byte 16 of the DECB if the status contains a unit check.
- Posts a completion code of hexadecimal 41 in byte 0 of the DECB.
- Returns control to the supervisor.

When the user does not request Error Recovery Procedures to be included in BTMOD, it is his responsibility to check the DECB. The user should check the status bytes (Figure 19). Generally, only one of these indicators properly describes the condition; the others indicate the secondary effects.

When a unit check occurs, the sense byte (Figure 20) is stored in byte 16 of the DECB. This byte should be checked on a unit check condition.

Error Counts

If the user wishes to keep error counts with his ERP on a line group basis, he must issue a LERB macro instruction to generate and initialize a Line Error Block and he must code ERLOGIC=C in the BTMOD macro

Bit	Condition
0	Attention
1	Status modifier
2	Control unit end
3	Busy
4	Channel end
5	Device end
6	Unit check
7	Unit exception
8	Program-controlled interruption
9	Incorrect length
10	Program check
11	Protection check
12	Channel data check
13	Channel control check
14	Interface control check
15	Chaining check

Figure 19. DECB Status Bytes

instruction. All error counts referring to a particular DTFBT are kept in one Line Error Block. The linkage between the DTFBT and the Line Error Block is indicated by the LERBADR operand in the DTFBT macro instruction.

If any of the error counters reach their threshold values before the transmission threshold value is reached, a message will be printed on the System/360 console typewriter. The message identifies the line and contains the three error counters and the transmission counter, together with their respective threshold values.

Bit	Explanation
0	Command reject
1	Intervention required
2	Bus out check
3	Equipment check
4	Data check
5	Overrun
6	Lost data
7	Timeout

Figure 20. DECB Sense Byte

After the message is printed, the transmission counter and the three error counters are added to a separate set of four cumulative counters and the counters are reset. If the transmission counter reaches its threshold before any of the error counts reach their thresholds, the four counters are added to the cumulative counters and the counters are reset, but no message is printed.

To print the cumulative counters, the user may issue a LERPRT macro instruction. When the macro instruction is issued, the current values of the error counters and the transmission counter are added to the set of four cumulative counters for the line. LERPRT may be used to print the cumulative counters for one line or for the entire line group. If the user does not want his cumulative counters to be set to zero after a LERPRT macro instruction, he must code CLEAR=NO in the macro instruction.

ERROR RECOVERY PROCEDURES FOR BINARY SYNCHRONOUS COMMUNICATION

The error recovery procedures (ERP) for binary synchronous communication are mandatory and are automatically included in the BTAM logic module. This is in contrast to the ERP for start-stop devices, which are optionally included when specified by the ERLOGIC operand in the BTMOD macro. Thus the ERLOGIC operand has no bearing on inclusion of binary synchronous communication ERP in the BTAM module. The only relevance of ERLOGIC to binary synchronous communication consists in the C and NC options which cause the Error Count Routines (common to both start-stop and binary synchronous communication) to be included in the BTAM module.

The purpose of these error recovery procedures is first to analyze status and sense information when I/O interruptions occur in conjunction with or as the result of errors in the initiation or execution of channel program commands. Then, if recovery of the error condition is possible, a retry procedure is attempted. The number of retries is specified by the user via the RETRY operand in the DTFBT macro instruction. When the error condition is of a class defined as irrecoverable or when the maximum number of retries fails to produce recovery, a system-to-operator message is provided, indicating the nature of the error. In addition, an error indication is posted in the appropriate DECB.

The entries in the following table refer to the actions taken by ERP when listed error conditions occur during the initiation or execution of the I/O commands.

Error Condition Command	Channel Data Check	Lost Data	Time Out	Inter-vention Required	Bus Out Check	Data Check	Overrun	Command Reject	Unit Ex-ception	Equip-ment Check
Set Mode	2	2	2	2	2	2	2	11	-	2
Enable	2	2	1	2	2	2	2	11	-	2
Dial	2	4	4	4	8	2	2	11	-	2
Disable	2	2	6	2	2	2	2	2	2	2
Prepare	2	2	2	7	2	2	2	11	-	2
Write	2	2	2	7	1	2	2	10	12	2
Read	2	3	5	7	2	9	9	11	-	2
Poll	2	2	2	7	2	2	2	11	12	2
Sense	1	1	1	1	1	1	1	1	1	1

In the description of the actions, n is the number of retries specified by the RETRY operand in the DTFBT macro instruction issued for the line group.

Action 1: Irrecoverable Error

A system-to-operator message is printed, indicating the nature of the error. The operation is posted complete with error (completion code: 41) and control is then returned to the problem program.

Action 2: Should-Not-Occur Error

The channel program is retried n times if the error occurs before data transfer. If n retries fail or if data transfer is taking place, Action 1 is performed. If data transfer is completed, the channel command is retried after the appropriate preliminary commands are executed (e.g., Write ENQ, Read response).

Action 3: Lost Data on Read

If the error occurs on a text-read and the last character received is not an end character, Read commands with counts of one are issued to read the ENQ character. After the ENQ character is received, a READ Repeat is issued. If n retries fail, Action 1 is performed.

If the last character received is an end character, a READ repeat is immediately issued. Action 1 is performed if n retries fail.

Action 4: Errors on Dial

If Lost Data or Intervention Required occur during a Dial command, a Disable command is issued and the dialing operation is retried. If n retries fail, Action 1 is performed.

If Time Out occurs on Dial, a Disable is issued and the operation is posted complete with a Time Out indication in the DECB. Since this condition merely means that no answer to the call was received, no retries are attempted and no error message is provided.

Action 5: Time Out on Read

If Time Out occurs on a text-read, Read commands with counts of two are tried. If ENQ or DLE, ENQ is received a READ Repeat is issued. If an EOT is received or if n consecutive retries fail, Action 1 is performed.

If Time Out occurs on a Read response, a WRITE inquiry is issued. If n retries fail, Action 1 is performed.

Action 6: Time Out on Disable

The Disable command is retried up to n times. If the error condition persists, Action 1 is performed.

Action 7: Intervention Required

If this error condition occurs on Prepare or Poll, a Disable command is issued. Then a Set Mode, Enable, and a retry of the channel program are executed. If the error condition persists, Action 1 is performed.

If the error condition occurs on Read or Write, Action 1 is performed.

Action 8: Bus Out Check on Dial

A DLE, EOT sequence is transmitted, a Disable and Set Mode are issued, and the Dial command is retried. If n retries fail to recover from the error condition, Action 1 is performed.

Action 9: Data Check or Overrun on Read

If the error occurs on a text-read, a READ repeat is issued. If n retries fail, Action 1 is performed. If the error occurs on a Read response, a WRITE Inquiry is issued. After n unsuccessful retries, Action 1 is performed.

Action 10: Command Reject on Write

An Enable command is issued and then Action 1 is performed.

Action 11: Command Reject

A DLE, ENQ sequence is transmitted, and then Action 1 is performed.

Action 12: Unit Exception on Write or Poll

A Read command with a maximum count and the skip flag on is executed and then Action 1 is performed.

In addition to the above actions, BTAM provides automatic retransmission of a message when a NAK response is received from the remote device. If n retries fail to recover, completion code 41 is set and bit 7 of byte 29 (error information field) is set to one.

If a message of the form:

STX--Text(may be zero length)--ENQ

is received, BTAM automatically responds by transmitting NAK to the remote device. If n retries fail to recover, BTAM performs Action 1. Completion code 41 is posted and bit 6 of the error information field (byte 29 of the DECB) is turned ON. Generally, n retries are not necessary, because the remote device sends EOT upon receiving the NAK. This is the standard procedure, for example, in communication with the 2780. The 2780 transmits a message of the above format in lieu of a record causing a card reader error or an internal buffer parity check. In this case, BTAM posts completion code 41, turns ON bit 6 of byte 29, and turns ON bit 1 (EOT) of the Flag byte.

Note: A message of the above format should not occur in CPU-to-CPU communication unless (user error) the ENQ character is transmitted within text in nontransparent mode.

STANDARD ERP FOR THE AUDIO RESPONSE UNITS

The following facilities are provided for the audio response units whether the error recovery procedures have been requested or not.

1. Hung up procedure: Whenever the unit exception bit is set in the status received at the end of an I/O operation for a 7770/7772 (indicating that the inquirer hung up during a transaction) and the unit check bit has not been set, the line is disconnected and one of the following procedures is followed:

    a. An initial operation, i.e., one starting with an Enable command (Read Initial, Write Initial and Write Invitational) is restarted and no completion posting is made.

    b. An operation specified with reset is posted normally at the completion of the reset function.

    c. If neither of the previous cases apply, the operation is posted with the I/O error completion code and the unit exception bit is set in the unit status byte field of the DECB.

2. Overlength procedure: Whenever BTAM detects that the inquirer has attempted to enter on a read operation an inquiry longer than specified in the length field of the DECB or MSGL field in the DTFBT, BTAM posts the operation with the I/O error completion code and the incorrect length indication bit set in the channel status byte field of the DECB. However, if there is an indication that the inquirer hung up, the overlength indication is disregarded.

BTAM PROGRAM CANCELLATIONS

BTAM may detect a program condition that makes it difficult or impossible to continue execution. In this situation, BTAM posts pertinent information in the cancel block and takes the proper action according to which BTAM logic routine is being executed.

The BTMOD cancel block is as follows:

Bytes	0	1	2	3
	Error Code	DTFBT Address		
	Reserved	DECB Address		
	4	5	6	7

If the BTAM interruption handler has control and a potential cancel condition is detected, the following actions are taken:

- The current operation is discontinued.

- A numeric error code, the DTFBT address, and the DECB address are posted to the BTMOD cancel block.

- A completion code of X'58' is posted to the ECB.

If any other BTAM routine has control and a potential cancel condition is detected, the following actions are taken:

- The execution of the BTAM routine which detected the cancel condition is discontinued.

- If the cancel condition has not been posted to the cancel block, a numeric error code, the DTFBT address, and the DECB address are posted to the cancel block.

- An operator awareness message is written on SYSLOG (see format and messages below).

- If CANCEL=YES was coded in the BTMOD macro instruction, a DOS CANCEL macro instruction is given.

- If CANCEL=NO was coded in the BTMOD macro instruction, a return code of X'01' is set in general register 15, the condition code is set to non-zero, and general register 1 points to the location containing the address of the BTMOD cancel block. Control is then returned to the user at the instruction following the BTAM macro instruction that caused the cancel condition to be detected.

The format of the operator awareness message is:

```
           {P}
[pp] 4BnnI {C}text DTFBT=aaaaaa,
DECB=aaaaaa
```

- pp for partition indicator, supplied by DOS when a multiprogramming supervisor is used.
- nn for the decimal error code.
- P indicates posting of information and return to user.
- C indicates a Cancel message.
- text for the explanation of the error.
- aaaaaa for the address in hexadecimal.

The BTAM cancel messages are:

Error Code	Text
00	User referred to closed DTFBT
01	DTFBT field improperly initialized
02	DECB field improperly initialized
03	Multiple wait count negative
04	Multiple wait count exceeds ECBLIST size
05	Attempt to process non-BTAM buffer
06	Unexpected program error in RELBUF
07	REQBUF count negative
08	RESETPL DECB and LCB DECB not same

Note: If either the DTFBT address or DECB address is not available or is not pertinent at the point at which the potential cancel condition is detected, the corresponding cancel block field will contain zero.

Programming Note: If the user who selected the BTMOD parameter CANCEL=NO elects to continue execution despite the error condition, he must store X'FF' in Byte 0 of the Cancel Block before returning control to BTAM.

ON-LINE TERMINAL TEST

The on-line terminal test facility is an optional service provided only if:

1. TERMTST=YES is coded in the BTMOD macro issued for the problem program.
2. TERMTST=YES is coded in the DTFBT macro issued to define the particular line group concerned.
3. TERMTST is coded in the TWAIT macro.

The on-line terminal test facility services test requests initiated at the remote device by either the operator or an IBM Customer Engineer. The test requests may be used as start-up procedures or for terminal checkout and terminal failure diagnosis. The tests provided operate on-line with the problem program and affect the user's problem program operation only to the extent that line time is required by the test facility to perform the requested functions over the selected line and CPU time necessary to set up the requested test.

Requesting of Tests

Specific test requests are initiated only at the remote device by the transmission to the CPU of a message having a special format. The tests requested may involve message switching, comparing of incoming data to a stored pattern in main storage, sending of a set of characters to a specified terminal, or checking of the IBM SELECTRIC ® typewriter print ball or carriage mechanisms on a remote terminal. Figure 21 shows the format of the message requesting a test. Start-of-text characters (viz, STX) are not shown in the format.

		FROM	TO	UNIT		END
99999	XX	TYPE	ADDR	SELECT	TEXT	CHARACTER

Figure 21. Format of Test Request Message

99999 (5 characters) - This is the primary action code used to recognize this message as a test request.

XX (2 characters) - This is the module sequence code used to define the specific test to be executed.

FROM TYPE (1 character) - Used by the terminal test modules to determine the type of terminal from which the test is being requested.

FROM TYPE	Device
space	2740C or 2740DC, when the 2760 OIU is to be tested.
1	1030, card reader

2	1050
3	1060
4	2740 (except 2760 OIU test)
5	1030, badge reader or manual entry
6	2260 (Remote)

TO
ADDR (1 or 2 characters) - This is the address of the terminal to which the message is to be sent.

TO ADDRess Usage:
This is a one byte field for the IBM 1030 Card Reader, 1050, 2740S, and 2740SC giving the addressing character for the selected terminal. For other 2740 combinations, the field is omitted.

The TO ADDR for 1030 badge readers and manual entry, 1060 devices, and 2260 devices is a 2 byte field. The address of the receiving terminal is selected by transmitting a predefined code in these character positions. The code is as follows:

1. 1060:

   01 will define terminal A as the receiving terminal.
   02 = terminal B
   03 = terminal C
   ⋮   ⋮
   26 = terminal Z

2. 1030 Badge Readers or Manual Entry

   02 will define terminal B as the receiving terminal.
   03 = terminal C
   04 = terminal D
   ⋮   ⋮
   26 = terminal Z

   Note: If 10 is entered as the address field, this will be considered an invalid request as the corresponding address (J) is not a legal 1030 address.

3. 2848 devices:

The TO ADDR is used to select the 2848 display control unit. The address of a display control unit can be any USASCII non-control character (i.e., any character in columns 3-7 in the USASCII code chart), therefore allowing 96 possible display control addresses.

Note: The TO ADDR applicable to a particular display control unit can be determined from one of its attached display stations by specifying the request address test in the test message.

Actual 2848 USASCII address	TO ADDR (2 bytes)
$b_7 \ldots b_1$	
0100000	01
0100001	02
⋮	⋮
1111111	96

UNIT
SELECT (0, 1 or 2 characters)

Unit select is not applicable to 1030 and 2740 devices; therefore, text can start in this position.

1. 1050 and 1060 devices (1 character):

   This specifies the particular unit at the TO ADDR that is to receive the message, i.e., 1052, 1053, 1055, 1062 Printer 1 or 2, etc. The appropriate unit select code can be determined from the publication related to the type of terminal being addressed.

2. 2260 (or 1053 attached to the 2848) devices (2 characters):

   2260 Display Stations and 1053 Printers are selected by transmitting a pre-defined code in these character positions. The device selection can be one of twenty-five USASCII non-control characters.

Actual Unit Address	UNIT SELECT (2 bytes)
$b_7 \ldots b_1$	
1000000	01
1000001	02
⋮	⋮
1011000	25

Note: The pre-defined 2-byte code applicable to a particular display station can be determined from that display station itself by utilizing the request address test.

END
CHARACTER

1030 = EOB
1050 = EOT
1060 = EOB
2740 = EOT
2848 = ETX

Note: The test message as transmitted from a 1060 device is entered by utilizing the data and transaction keys. The EOB character is entered by depressing the teller A or B key.

Applicable Module Sequence Codes for 1030, 1050, 1060, 2740 and 2260:

01 Message Switching

This module receives a message from the requesting terminal and transmits it to the terminal (on the same line) as specified in the test message. Note: The number of characters that can be switched is directly dependent upon the count field that the user specified in the problem program READ macro for the line over which the test is requested. Only this number of characters can be switched.

02 Tilt

This module sends the tilt test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

03 Rotate

This module sends the rotate test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

04 Twist

This module sends the twist test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

05 Stored Compare

This module provides a means to compare the received message with a particular pattern stored in the CPU. The defined message in main storage is compatible with the transmitting capabilities of the terminal(s) involved.

The test message to be compared with the stored pattern is transmitted from the terminal and consists of the numbers 0 through 9 followed by the alphabet (A through Z). The incoming test message must specify the comparison characters in the same order although not all of them need be specified.

The number of characters that can be compared is directly dependent upon the count field that the user specified in the problem program READ macro for the line. The I/O area must be long enough to contain the header information (99999,etc.), the characters to be compared, and the end character.

Exceptions:

1. When transmitting from a 2740 terminal with Station Control a space character must precede the comparison data. For the Basic or 2740 with Transmit Control, two space characters are needed.

2. The comparison test for a 1060 is requested by entering the following message:

999996534210	TELLER A or B

Comparison is then made to this message. Responses to this request are printed only at the requesting terminal.

Messages Received at the terminal are:

1. If the comparison to the stored message is valid, the following message is sent to the terminal specified in the TO ADDR field:

CMP VLD-*

The character printed in the position of the asterisk will be the last character against which a comparison could be made. Exception: The message sent to a 1060 after a valid comparison is:

CMP VLD

If the request was received properly, but an insufficient count was specified in the READ and thus no characters could be compared, a / character is printed in the asterisk position.

2. If the comparison to the stored message is invalid, the data received is merely message switched to the terminal specified in the TO ADDR.

Note: The stored compare test is not applicable to the 1030 manual entry or badge reader.

06 All Characters Test

This module provides a standard all-characters test for C E terminal checkout and serves as a start-up message for the customer. Special characters are not used in the terminal test. Characters received at the terminal are:

1. 1030, 1060, 2848 (2260 and 1053) Numbers: 0-9, and alphabet: A-Z.

2. 1050, 2740 Numbers: 0-9, alphabet: A-Z (lower case), and alphabet A-Z (upper case).

07 SELECTRIC Analyzer Test

This module provides an exercise to analyze the capability of the SELECTRIC typewriter carriage mechanism to perform within defined specifications. The defined message in storage, when transmitted, will have the ability to so exercise a requested terminal. This test is not applicable to a 1053 Printer attached to a remote 2848 Control Unit.

08 Write at Line Address Test (2260 only)

This test will provide a line selectivity test by using the first two characters after the UNIT SELECT field as a new line code. This can be followed by data which is to be switched to the terminal and written on the line specified on the display station screen. The following characters are used to select the line on the display station screen:

01 = line #1

02 = line #2

03 = line #3

12 = line #12

09 Request Address Test (2260 only)

The TO ADDR and UNIT SELECT fields are not utilized in this test message since the test itself provides these fields to the requesting terminal. ETX can be sent immediately after the TYPE field. The 9-byte message returned to the requesting display station is in the following format:

D	C	+	D	V	DC Addr	DV Addr

DC address is the pre-defined code necessary to select this display control unit (2 bytes). DV address is the pre-defined code necessary to select this display station (2 bytes).

Note: This test provides the TO ADDR and UNIT SELECT codes of the requesting 2260 itself. It is not a means of getting the TO ADDR and UNIT SELECT codes for some other 2260.

10 2760 Optical Image Unit Frame Change Test (2740C, 2740DC)

11 2760 Optical Image Unit Scan Point Test (2740C, 2740DC)

These tests are requested by the following message being entered through the Selectric:

99999xx(SP)(BS)(F)$(A_1)$$(A_2)$ Ⓑ

The meaning of the variable quantities is as follows:

xx	=	10 for frame change, 11 for scan point
(SP)	=	space character (blank)
(BS)	=	backspace character
(F)	=	function control character
(A_1), (A_2)	=	code characters for the amount of film motion

Note: See Functional Specifications of the device for the values of (F), (A_1), and (A_2). On receipt of the test request BTAM writes a Ⓨ to the terminal and issues another Read. The engineer must then key a Ⓒ to activate the test.

BTAM next writes a frame-change message derived from the test request, namely Ⓓ (PRE)(O)(F)$(A_1)$$(A_2)$ Ⓑ.

BTAM reads the 2760 response to this message. If this is Ⓨ, BTAM writes Ⓒ to activate the 2760 OIU. If the test is a frame change test, BTAM restarts the original channel program at this point.

To continue with a scan point test, BTAM issues another Read to the 2760 OIU. The engineer may then enter one pair of coordinates with the light-pen if in Auto EOM mode, or several pairs if in Manual EOM mode (followed by Ⓑ for end entry). On completion of this input, BTAM writes a Ⓨ to the 2760 OIU and reads in the resulting (automatically generated) Ⓒ. BTAM then writes Ⓓ and one idle character to the 2740 printer, followed by the data read from the 2760 OIU. BTAM reads the 2740 response to this message. If this is Ⓨ, BTAM writes Ⓒ to the terminal and restarts the original channel program.

Terminal Test Restrictions

1. The problem program I/O area must be of sufficient length so that the entire test message can be read into main storage. If buffers are used, there is an additional restriction: the data area of the first buffer in the chain must contain all of the characters in the test request message through the UNIT SELECT field. In the case of the write-at-line-address test request the two characters defining the line address must also fit into the first buffer. The TEXT characters and END CHARACTER may also appear in the first buffer.

2. If the problem program is utilizing an answer list, none of the reset options can be specified in the READ macros. The line connection must remain established for the terminal tests.

3. To request a test from a 1030 Badge Reader, the badge reader must be wired to read out the entire 10 columns of the badge (refer to 1030 publications).

4. The transaction code received from 1030 devices is not included as part of the test request.

5. All 1030 tests require a 1033 Printer on the same line as the requesting terminal. The printer is specified in the TO ADDR field.

6. The terminal tests will not test 1035 Badge Readers or 1030 Badge Readers in a 1035 environment.

ON-LINE TERMINAL TEST FOR BINARY SYNCHRONOUS COMMUNICATION

The on-line terminal test for Binary Synchronous Communication (BSC) is an optional service provided by DOS/BTAM. It is provided to assure the proper operation of the system, and it may be used in the diagnosis and correction of a BSC system malfunction.

The on-line test facility will communicate between S360 and S360, S360 and 1130, or S360 and 2780. For S360 to S360 operation, both computers may have DOS/BTAM or OS/BTAM with the on-line test facility, or one computer may have BTAM and the other an on-line diagnostic. For S360 to 1130 operation, the S360 must have BTAM, and the 1130 must have an on-line diagnostic. Operation between S360 and 2780 requires BTAM in the S360. The 2780 is not a stored program device.

In communication between S360 and S360, either CPU may initiate BSC on-line terminal test. In communication between S360 and 2780, or S360 and 1130, the S360 can not initiate BSC on-line terminal test, except when X=0. When X=0, any BSC device (1130, 2780, or S360) may initiate on-line test. When the S360 initiates BSC on-line terminal test with a 2780, the mode switch at the 2780 must be set to either PRINT or PUNCH, if the 2780 is on a point-to-point line.

The BSC on-line terminal test is provided on two levels. First: DOS/BTAM recognizes Request-for-Test (RFT) messages transmitted by the remote unit. When a RFT message is recognized, BTAM performs the requested test, which is usually transmission of a test message. Second: DOS/BTAM initiates on-line test by transmitting an RFT message requesting the remote unit to perform a test that is specified by the parameters of the ONLTST macro.

The BSC on-line test facility recognizes Request-for-Test messages only if:

1. The operation is a Read Initial.
2. BSCTEST=YES is coded in the BTMOD macro being used by the problem program.
3. TERMTST=YES is coded in the DTFBT macro issued to define the particular line group concerned.

The BSC on-line test facility services a RFT message only if:

1. The Request-for-Test is received without error.
2. The problem program issued a TWAIT following the Read Initial macro instruction.

Note: The problem must insure that the count in the Read Initial operation is large enough to accommodate the RFT message and any test message that is sent to the remote unit. The largest RFT message allowed is 300 bytes. Buffers may be used, but the first buffer should be large enough to accomodate the RFT message any any test message.

In most cases response to an RFT message consists of transmitting a fixed test message to the remote unit a specified number of times. In these cases the Interruption Handler turns on the terminal test flag in the DECB and posts the DECB complete without error. The on-line terminal test logic gets control through TWAIT. The requested test pattern is loaded into the problem program I/O area and transmitted to the remote unit. In one special case (when X=0), the test message is only acknowledged When the required number of transmissions have been completed, an EOT is transmitted on non-switched lines, or a DLE EOT is transmitted on switches lines, and a Disable command is issued. The results of the test are printed on the computer console, and the problem program operation is restarted.

When BTAM initiates a binary synchronous on-line test procedure, the following sequence of events occurs:

1) The problem program issues an ONLTST macro instruction which generates linkage to the on-line test logic generated by the BTMOD macro instruction.

2) The on-line test logic uses the parameters supplied by the ONLTST macro to generate an RFT message.

3) The RFT message is transmitted, and test messages from the remote unit are read into the area supplied by the on-line test module.

4) The results of the completion of the test are printed on the computer console. The completion code in the DECB is posted X'7F' if the test ends normally, and X'41' otherwise.

5) The problem program may wait for the completion of the ONLTST macro through the TWAIT or WAIT macro. Normal operation of all other lines the the problem program will continue without interruption during the operation of the on-line terminal test.

Two types of messages are used by the BSC on-line test facility:

1. The RFT message is used to request that on-line test procedures be initiated. Its format includes various identification parameters and may include text characters.

2. The test message consists only of text characters and the appropriate framing characters. Appropriate framing characters are STX and ETX for normal text and DLE STX and DLE ETX for transparent datal

The format of the RFT message is:

S O % H	X	Y	N	TI	[D L E]	S T X	[TEXT]	[D L E]	E T X

S
O % (2 characters) - This identifies the
H message as an RFT message

X (2 characters) - This is a two-byte zoned decimal field which defines the type of test to be performed.

Y (2 characters) - This is a two-byte decimal field which defines the number of times the test message is to be transmitted. If X=0 and the remote terminal is a 2780, Y must equal 1 because only the RFT message will be transmitted by the 2780.

N (1 character) - This is a one-byte decimal digit (0-9) which specifies the size of the TI field. If N≠0, then the TI field must be present. For point-to-point lines, N must equal zero.

TI (0-9 characters) - This is the selection address of the unit to which the test message will be sent. This unit need not be the same unit that sent the RFT message. The TI field may specify another unit on the same line as the unit that transmitted the RFT message. This field is not present if N=0.

[DLE] STX (1 or 2 characters) - These are framing characters for start of text in the RFT message. DLE is present if text is to be written in transparent mode.

TEXT - This field is optional and appears only when X=1.

ETX - This is the framing character for end of text in the RFT message for non-transparent data. For transparent data, BTAM provides the end characters.

Note: When intermediate block check mode is used, an error information byte will be present in 360 core.

Note: If an RFT message is transmitted from a 2780 with the Auto Turnaround Feature, and the terminal test requester at the 2780 wishes to turn the line around after the transmission, blank cards should follow the RFT message with the ETX character deleted from the RFT message. This allows the Auto Turnaround Feature to be utilized so that the terminal test message can be received at the punch of the 2780. This procedure can not be used when X=0 or X=1 in the RFT message.

The contents of the test message used for BSC on-line test are defined by the X field of the RFT message. Allowable values of X are 00-19, and they have the following meanings.

X=00 The test message is to be acknowl edged ACK if received without errors. It is acknowledged NAK if a data check is detected, and it is not responded to at all, if other ending conditions are detected.

X=01 The text received with the RFT message will be transmitted Y times.

X=02 Transparent EBCDIC Message:

```
DLE STX text DLE ETX
```

The text consists of all 256 EBCDIC characters in collating sequence order.

X=03 USASCII transparency message:

```
DLE STX text DLE ETX
```

The text is in USASCII code (high order bit always zero), and consists of all 128 USASCII characters in collating sequence order.

X=04 Normal EBCDIC Message:

```
STX text ETX
```

The text consists of the 245 non-data link control characters. The characters excluded are SOH, STX, ETX, ETB, EOT, ENQ, ACK, NAK, SYN, US, DLE.

X=05 Normal USASCII Message:

```
STX text ETX
```

The text is in USASCII code, and consists of the 117 non-data link control characters. The excluded characters are the same as X=04.

X=06 Alphameric USASCII Message:

```
STX A B C D E F G H I J K L M N O P
Q R S T U V W X Y Z 0 1 2 3 4 5 6 7
8 9 ETX
```

X=07 USASCII printer message:

```
STX ESC Q A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z 0 1 2 3 4
5 6 7 8 9 ETX
```

This message is used for the 2780 printer.

X=08 USASCII punch message:

```
STX ESC 4 A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z 0 1 2 3 4
5 6 7 9 ETX
```

This message is used for the 2780 punch.

X=09 TRANSCODE printer message:

```
STX ESC / A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z 0 1 2 3 4
5 6 7 8 9 ETX
```

This message is coded in TRANSCODE and is used for the 2780 printer.

X=10 TRANSCODE punch message:

```
STX ESC 4 A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z 0 1 2 3 4
5 6 7 8 9 ETX
```

This message is coded in TRANSCODE and is used for the 2780 punch.

X=11 TRANSCODE multi-point message:

```
STX A B C D E F G H I J K L M N O P
Q R S T U V W X Y Z 0 1 2 3 4 5 6 7
8 9 ETX
```

This message is coded in TRANSCODE and may be used for either the 2780 punch or printer.

X=12 EBCDIC printer message:

This message has the same content as the TRANSCODE printer message, X=09. The code is EBCDIC. This message is used for the 2780 printer.

X=13 EBCDIC punch message:

This message has the same content as the TRANSCODE punch message, X=10. The code is EBCDIC. This message is used for the 2780 punch.

X=14 EBCDIC alphameric message:

This message has the same content as the TRANSCODE multi-point message, X=11. The code is EBCDIC, and it may be used for the 2780 printer or punch on multipoint lines.

X=15 EBCDIC Weak Pattern Message (Switched Line):

The text consists of 74 bytes of X'00 X'00' followed by 6 bytes of SYN.

X=16 EBCDIC Weak Pattern Message (Leased Line):

The text consists of 40 bytes of X'AA' followed by 40 bytes of X'55'

X=17 TRANSCODE Weak Pattern Message (Switched Line):

The text is 80 SOH characters.

X=18 TRANSCODE Weak Pattern Message (Leased Line):

The text is 40 N's followed by 40 ESC characters.

X=19 EBCDIC Weak Pattern (DLE SYN insertion):

The text is 280 bytes of X'00' followed by 10 SYN characters. This message is transmitted in transparent code.

The on-line test facility prints the results of the test on the computer console. Two messages are provided; one is used when BTAM is transmitting test messages (or RFT messages with X=0), and the other is used when BTAM is receiving test messages. The content of these messages is:

Transmitter	Receiver
Line Address	Line Addresss
Number of Transmissions (Y)	Number of Transmissions
X Field	X Field
Time Outs	Time Outs
NAK's Received	Lost Data Occurrences
Terminal ID (multipoint)	Data Checks

The formats of these messages are:

For the transmitter:

```
4B70I  ON-LINE TEST cuu xx yy TO NK TI
```

For the receiver:

```
4B71I  ON-LINE TEST cuu xx yy TO LD DC
```

Where:

4B70I identifies the message as BSC on-line test results for the transmitter.

4B71I identifies the message as BSC on-line test results for the receiver.

cuu
Specifies the line in the form channel and unit.

xx
Specifies the test type. This is the X field from the RFT message.

yy
Specifies the number of transmissions. For the transmitter, this value is the value from the RFT message. For the receiver, this value is accumulated by the On-Line test program.

TO

Specifies the number of occurrences of time out.

NK

Specifies the number of NAK's received by the transmitter.

TI

Specifies the terminal ID. This is printed only on multi-point lines.

LD

Specifies the number of occurrences of lost data.

DC

Specifies the number of occurrences of data check.

WESTERN UNION PLAN 115A TERMINALS

Terminal To CPU

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST} ,({xxyy-hexchars,}...) {WRAPLST}

Each terminal on a line is invited to send with a two-character code. The first character is always an 'X'. The second character identifies the terminal. The four characters xxyy are the hexadecimal representation of the invitation to send code as it appears in shifted Baudot code (see Appendix B). Thus, xx is always 17; yy is the letter that selects the station. The list may be an open list or a wrap-around list.

READ Initial (TI): The problem program issues a READ initial macro instruction to start or restart polling.

The channel program executed is:

(1)	Write	EOT sequence	Note 1
(2)	Write	invitation code	Note 2
(3)	Read	Response	Note 3
(4)	Read	Data	Note 4

Note 1: The EOT sequence (↑H↓↓↓↓↓↓↓) is sent to put all terminals on the line in control mode. (↑ = upshift or FIGS; ↓= downshift or LTRS)

Note 2: The invitation to send code is sent to poll the first (or next) terminal in the terminal list.

Note 3: The response to polling is read into the first two bytes of the input area. If the response is negative (single character V or M), the channel program is interrupted because the single character gives a wrong-length indication. Command chaining is suppressed. BTAM recognizes the interruption, initializes the channel program, and restarts the channel program at command (2) to poll the next terminal. If a negative response occurs at the end of an open list, the operation is posted as complete in the ECB. A positive response to polling is the message itself. Command (3) then chains to Command (4).

Note 4: The message is read until the EOT sequence is received or the count reaches zero. The operation is posted as complete.

CPU To Terminal

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars,}...)

Each terminal on a line is addressed by a two-character call code. The first character is the circuit call. The second character identifies the terminal. Normally the addressing list contains a single call code. A message can be sent to multiple terminals on a line (see Note 3). The four characters xxyy are the hexadecimal representation of the call code as it appears in shifted Baudot code (see Appendix B). Thus xx is the circuit call and yy is the station identifier.

WRITE Initial (TI): A WRITE initial macro instruction is used to send a message to one or several stations. The line may be reset after the message is sent. The channel program for WRITE TI is:

(1)	Write	EOT sequence	Note 1
(2)	Write	Call code	Note 2
(3)	Read	Response	Note 3
(4)	Write	Data	Note 4

Note 1: The EOT sequence is sent to put all terminals on the line in control mode.

Note 2: The call code is sent to address the first (or next) terminal in the terminal list.

Note 3: The response to addressing is read into the first byte of the DECB response field. If the response is negative (no response), the operation is posted as complete-with-error in the ECB. If the response is positive (single character V or M) and the end of the list has not been reached, BTAM restarts the channel program at command (2) to address the next terminal. If the end of the list has been reached, command (3) chains to command (4).

Note 4: The output message must begin with an end-of-addressing (EOA) character. For the 115A it is a space character. The length specified in the WRITE macro instruction must be exact. The last character of the message should end with an EOT.

Terminal to Terminal

BTAM does not support terminal-to-terminal traffic between terminals on a line.

WRITE Break (TB): The WRITE Break macro instruction is not used for normal message processing. It should be issued when a data check occurs, due to terminal-to-terminal traffic. It is used to stop a terminal which is sending data. A single break CCW causes the transmission control unit to send continuous space signals to the line adapter. The length operand specifies the number of space signals to be sent. The area and entry operands are not required in this type of WRITE macro.

AT&T 83B3 SELECTIVE CALLING TERMINALS

Terminal to CPU

Name	Operation	Operand
symbol	DFTRMSLT	{OPENLST} {WRAPLST}, ({xxyy-hexchars,}...)

Each terminal on a line has a unique two-character transmitter start code (TSC). Normally all of the terminals on a line are polled with a single terminal list. The list may be either an open list or a wrap-around list. The four characters xxyy are the hexadecimal representation of the TSC as it appears in the shifted Baudot code (see Appendix B).

READ Initial (TI): The problem program issues a READ initial macro instruction to start or restart polling. The channel program executed is:

(1)	Write	EOT sequence	Note 1
(2)	Write	TSC	Note 2
(3)	Read	Response	Note 3
(4)	Read	Data	Note 4

Note 1: The ↑H↓ sequence is sent to put all terminals on the line in control mode. (↑ = FIGS; ↓ = LTRS).

Note 2: The transmitter start code is sent to poll the first (or next) terminal in the terminal list.

Note 3: The response is read into the first two bytes of the input area. If the response is negative (single character V or M), BTAM restarts the channel program at command (2) to poll the next terminal. If the end of an open list is reached before a positive response is received, the operation is posted as complete in the ECB. If the response is positive (the message itself), command (3) data chains to command (4).

Note 4: The message is read until an EOT is received or the count reaches zero. The operation is posted as complete in the ECB.

CPU to Terminal

Name	Operation	Operands
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars,}...)

Each terminal on a line has a unique two-character call directing code (CDC). Normally the terminal list contains a single CDC. Messages can be sent to multiple terminals on a line. (See Note 4 below.) The four characters xxyy are the hexadecimal representation of the CDC as it appears in shifted Baudot code (see Appendix B).

WRITE Initial (TI): A WRITE initial macro instruction is used to send a message to one or several stations. The line may be reset after the message is sent. The channel program for WRITE TI is:

(1)	Write	EOT sequence	Note 1
(2)	Write	CDC	Note 2
(3)	Write	LTRS	Note 3
(4)	Read	Response	Note 4
(5)	Write	Data	Note 5

Note 1: The EOT character set (FIGS, H, LTRS) is sent to put all of the terminals on the line in control mode.

Note 2: The call directing code (CDC) is sent to address the first (or next) terminal in the terminal list.

Note 3: The LTRS character is sent to trigger the response from the terminal.

Note 4: The response to addressing is read into the first two bytes of the input area. If any terminal in the list fail to answer (negative response), the operation is posted as complete-with-error in the ECB. If the response is positive (single character V or M) and the end of the list has not been reached, BTAM restarts the channel program at command (2) to address the next terminal. When the end of the list is reached, command (4) chains to command (5).

Note 5: The output message must begin with an EOA character set (carriage return (CR), line feed (LF), LTRS). The length specified in the WRITE macro instruction must be exact. The last character of the message should be the EOT sequence.

Terminal to Terminal

BTAM does not support terminal-to-terminal traffic between terminals on a line.

WRITE Break (TB): The WRITE Break macro instruction is not used for normal message processing. It should be issued when a data check occurs, due to terminal-to-terminal traffic. It is used to stop a terminal which is sending data. A single break CCW causes the transmission control unit (TCU) to send continuous space signals to the line adapter. The length operand specifies the number of space signals to be sent. The area and entry operands are not required in this type of WRITE macro.

IBM 1030 TERMINALS

Terminal to CPU

Name	Operation	Operand
symbol	DTFRMLST	{OPENLST}, {WRAPLST} ({xx-hexchars,}...)
symbol	DFTRMLST	{SSALST }, {SSAWLST} {xx-hexchars,}...

Each IBM 1031 Input Station on a line is polled with one polling character. The two characters xx are the hexadecimal representation of the terminal polling character, as it appears in the six-bit BCD transmission code (see Appendix B). The list may be an open list, a wrap-around list, or an Auto Poll list.

READ Initial (TI) and READ Initial with Reset (TIR) using OPENLST or WRAPLST: A READ initial macro instruction is used to start or restart polling on a line. The channel program for READ TI using OPENLST or WRAPLST is:

(1)	Write	EOT sequence	Note 1
(2)	Write	Polling character	Note 2
(3)	Read	Response	Note 3
(4)	Read	Data	Note 4
(5)	Write	ⓎⒸⒸⒸ (TIR only)	Note 5

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The polling character to poll an entry in the list is sent.

Note 3: The response to polling is read into the first two bytes of the input area. If the response is negative (single character Ⓝ), BTAM re-initializes the channel program at command (2) to poll the next nonskipped entry in the list. If the end of an open list is reached before a positive response is received, the operation is posted as complete in the ECB. If the response is positive (Ⓓ) and the second character of text received is not an EOB or EOT, command (3) data chains to command (4) to read the remaining bytes of text. If the second character received is an EOB or EOT, the operation is posted as complete in the ECB.

Note 4: The message is read until an EOB is received. The operation is then posted as complete in the ECB.

If READ TIR was specified and no error occurred during transmission, command (4) chains to command (5). Otherwise, the channel program is terminated.

Note 5: Ⓨ and EOT sequence are sent to acknowledge the message and to reset the line.

READ Initial (TI) and READ Initial with Reset (TIR) Using Start-Stop Auto Poll List: The channel program using an open start-stop Auto Poll list (SSALST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	NOP	Note 3
(4)	Read response	Note 4
(5)	Read message	Note 5
(6)	Write ⓎⒸⒸⒸ (TIR only)	Note 6

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), halting the channel program.

Note 3: This is the last command executed when all negative responses are received.

Note 4: This command is executed on a positive response to bring the index byte (corresponding to the responding terminal) and one text byte into the input area.

Note 5: The message is read into the input area until an EOB is received. The operation is posted as complete in the ECB. If READ TIR is specified and no error occurred during transmission, command (5) chains to command (6). Otherwise, the channel program is terminated.

Note 6: On a successful poll Ⓨ is sent to indicate that the read operation occurred without error. The ⒸⒸⒸ turns the terminal motors off.

READ Initial (TI) and READ Initial with Reset (TIR) using Start-Stop Wraparound Auto Poll List: The channel program using a start-stop wraparound Auto Poll list (SSAWLST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	TIC (5)	Note 3
(4)	TIC (7)	Note 4
(5)	Poll first-entry	Note 5
(6)	TIC (5)	Note 6
(7)	Read response	Note 7
(8)	Read text	
(9)	Write ⓎⒸⒸⒸ (TIR only)	

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which restarts polling.

Note 3: On a negative response to polling, at the end of the list, this TIC command is executed to start the second Poll command.

Note 4: On a positive response to polling, this TIC command is executed to start the READ response command.

Note 5: If either Poll command terminates with a negative response at the end of the list, this command restarts polling at the beginning of the list.

Note 6: On a negative response to polling at the end of the list, this TIC command is executed to restart Poll command (5).

Note 7: Commands (7)-(9) operate the same as commands (4)-(6) of READ initial using SSALST.

READ Continue (TT) and READ Continue with Reset (TTR): A READ continue macro instruction is issued after a successful READ TI to send a positive response and chain to a READ initial sequence to restart polling. The channel program for READ TT is:

(1) Write Response and EOT Note 1
(2) Commands (2) through (5) are the same as commands (2) through (5) of a READ initial.

Note 1: A positive response (Ⓨ) followed by ⒸⒸⒸ , is sent over the line.

READ Repeat (TP) and READ Repeat with Reset (TPR): A READ repeat macro instruction is issued after a non-successful READ TI, TT or TP to send a negative response and chain to a READ initial sequence to restart polling. The channel program for READ TP is:

(1) Write Response and EOT Note 1
(2) Commands (2) through (5) are the same as commands (2) through (5) of READ TI.

Note 1: A negative response (Ⓝ) followed by ⒸⒸⒸ , is sent over the line.

READ Skip (TS): The READ skip macro instruction generates a read command with the skip bit on. This macro allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by the user's count. The channel program for READ TS is a single read skip CCW.

CPU to Terminal

Name	Operation	Operand
symbol	DFTRMLST	OPENLST,(xx-hexchars)

Each 1033 printer on a line is addressed with a single character code. An addressing terminal list may contain only one printer address. The two characters xx are the hexadecimal representation of the addressing character, as it appears in the six-bit BCD transmission code (see Appendix B). The list must be defined as an open list.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): A WRITE initial macro instruction is issued to establish connection with the printer and send a message block to it. The line may also be reset after the message is sent. The channel program for WRITE TI and WRITE TIR is:

(1)	Write	ⒸⒸⒸⓈ	Note 1
(2)	Write	Addressing character	Note 2
(3)	Write	"1"	Note 3
(4)	Read	Response	Note 4
(5)	Write	Message block	Note 5
(6)	Read	Response	Note 6
(7)	Write	ⒸⒸⒸ (TIR only)	Note 7

Note 1: A character sequence is sent to condition the 1031 to receive the printer addressing character.

Note 2: The addressing character is sent, just as it appears in the terminal list.

Note 3: A single character '1' is sent to prepare the 1031 to send the response character.

Note 4: The response is read into the first byte of the DECB response field. If the response is negative (Ⓝ), the operation is posted as complete in the ECB. If the response is positive (Ⓨ), command (4) chains to command (5) to send the message.

Note 5: The message is sent. Each outgoing message block must begin with Ⓓ and end with Ⓑ . Write mark characters (DF in hexadecimal representation) must be present between all message and/or control characters. A three-character delay is used between the output printing characters. Additional delay is required for carriage return, line feed, tabs, etc.

Note 6: The response to vertical redundancy check (VRC) is received. If the response is positive and TIR is specified, command (6) chains to command (7).

Note 7: An EOT sequence is sent to set the terminal in control mode.

WRITE Continue (TT) and WRITE Continue with Reset (TTR): A WRITE continue macro instruction is issued to send additional message blocks to the same printer. The reset option may also be specified. The channel program is the same as steps 5, 6, and 7 shown for WRITE TI and TIR. Note, however, that the first character of these successive message blocks is a text character, not Ⓓ as with the first block. No terminal list is used with this macro instruction.

WRITE Positive Acknowledge (TA): A WRITE positive acknowledge macro instruction is issued by the problem program after a successful READ operation to send a positive response and to stop line activity. The channel program for a WRITE TA is:

(1) Write ⓎⒸⒸⒸ

WRITE Negative Acknowledge (TN): A WRITE negative acknowledge macro instruction is issued by the problem program after a WRITE operation or a non-successful READ operation to send a negative response and to stop line activity. The channel program for WRITE TN is:

(1) Write ⒸⒸⒸ

IBM 1050 TERMINALS ON NONSWITCHED NETWORKS

Terminal to CPU

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST}, {WRAPLST} ({xxyy-hexchars,}...)
symbol	DFTRMLST	{SSALST }, {SSAWLST} {xxyy-hexchars,}...

Each component on a line is polled with a two-character code. The first of these two polling characters identifies the terminal. The second either identifies a single component of that terminal or is the common polling character (0). The four characters xxyy are the hexadecimal representation of the polling characters as they appear in the six-bit BCD transmission code (see Appendix B). Thus, xx identifies the terminal and yy the component. The list may be an open list, a wrap-around list, or an Auto Poll list.

<u>READ Initial (TI) and READ Initial with Reset (TIR) using OPENLST or WRAPLST:</u> A READ initial macro instruction is used to start or restart polling on a line. The channel program for READ TI using OPENLST or WRAPLST is:

(1)	Write	EOT sequence	Note 1
(2)	Write	Polling characters	Note 2
(3)	Read	Response	Note 3
(4)	Read	Data	Note 4
(5)	Write	ⒹⒸⒸⒸ (TIR only)	Note 5

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to set all of the terminals on the line in control mode.

Note 2: Polling characters are sent to poll an entry in the list.

Note 3: The response to polling is read into the first two bytes of the input area. If the response is negative (only one character received), BTAM re-initializes the channel program at command (2) to poll the next entry in the list.

If the end of an open list is reached before a positive response is received, the operation is posted as complete in the ECB. If the response is positive (the response character and first byte of text are read), command (3) chains to command (4).

Note 4: The remaining bytes of text are read until an EOB is received. Completion of the operation is then posted in the ECB.

If READ TIR was specified and no error occured during transmission, command (4) chains to command (5). Otherwise, the channel program is terminated.

Note 5: Ⓓ and EOT sequence are sent to acknowledge the message and to reset the line.

<u>READ Initial (TI) and READ Initial with Reset (TIR) Using Start-Stop Auto Poll List:</u> The channel program using an open start-stop Auto Poll list (SSALST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	NOP	Note 3
(4)	Read response	Note 4
(5)	Read message	Note 5
(6)	WriteⒹⒸⒸⒸ (TIR only)	Note 6

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which halts the channel program.

Note 3: This is the last command executed when all negative responses are received.

Note 4: This command is executed on a positive response to bring the index byte (corresponding to the responding terminal) and one text byte into the input area.

Note 5: The message is read into the input area until an EOB is received. The operation is posted as complete in the ECB. If READ TIR is specified and no error occurred during transmission, command (5) chains to command (6). Otherwise, the channel program is terminated.

Note 6: On a successful poll, Ⓓ is sent to indicate that the read operation occurred without error. The ⒸⒸⒸ turns the terminal motors off.

READ Initial (TI) and READ Initial with Reset (TIR) using Start-Stop Wraparound Auto Poll List: The channel program using a start-stop wraparound Auto Poll list (SSAWLST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	TIC (5)	Note 3
(4)	TIC (7)	Note 4
(5)	Poll first-entry	Note 5
(6)	TIC (5)	Note 6
(7)	Read Response	Note 7
(8)	Read text	
(9)	Write ⒹⒸⒸⒸ (TIR only)	

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which resets polling.

Note 3: On a negative response to polling, at the end of the list, this TIC command is executed to start the second Poll command.

Note 4: On a positive response to polling, this TIC command is executed to start the Read response command.

Note 5: If either Poll command terminates with a negative response at the end of the list, this command restarts polling at the beginning of the list.

Note 6: On a negative response to polling at the end of the list, this TIC command is executed to restart Poll command (5).

Note 7: Commands (7)-(9) operate the same as commands (4)-(6) of READ initial using SSALST.

READ Continue (TT) and READ Continue with Reset (TTR): A READ continue macro instruction is used after a successful READ TI, TT, or TP to read subsequent blocks of data from the same component, without repolling. The channel program writes a positive response Ⓨ and chains to commands (4) and (5) of READ initial.

READ Repeat (TP) and READ Repeat with Reset (TPR): A READ repeat macro instruction is issued following an unsuccessful read operation to reread the last message from the same component into the same storage area as in the previous operation. The channel program is the same as READ TT, except that it writes Ⓝ instead of Ⓨ. For paper tape or card input, READ TP requires the line correction feature. For these components, READ TP can be used only twice in succession, for a total of three tries. Furthermore, it can be used with the 1054 Paper Tape Reader only for message blocks that are equal to or less than 312 characters in length.

READ Skip (TS): The READ skip macro instruction generates a read command with the skip bit on. This macro allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by a user-supplied count. The channel program for READ TS is a single read skip CCW.

CPU to Terminal

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars,}...)

Each terminal on a line is addressed with a two-character code. The first of these two addressing characters identifies the terminal. The second selects one or all of the components of that terminal and conditions the component(s) to receive a message. A terminal list may contain more than one component address. The four characters xxyy are the hexadecimal representation of the addressing characters as they appear in the six-bit BCD transmission code (see Appendix B). Thus, xx identifies the terminal and must be the same for all entries in the list and yy identifies the component.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): A WRITE initial macro instruction is issued to address one or more components of a terminal and send a message block. The terminal may be reset if desired. The channel program for WRITE TI and WRITE TIR is:

(1)	Write	EOT sequence	Note 1
(2)	Write	Addressing characters	Note 2
(3)	Read	Response	Note 3
(4)	Write	Ⓓ	Note 4
(5)	Write	Message block	Note 5
(6)	Read	Response	Note 6
(7)	Write	EOT sequence (TIR only)	Note 7

Note 1: An EOT sequence (ⒸⒸⒸ) is sent to set all terminals on the line in control mode.

Note 2: The addressing characters are sent to address a device.

Note 3: The response to addressing is read into the first byte of the DECB response field. A negative response from any component causes the operation to be posted as complete-with-error in the ECB. On positive response, the channel program is re-initialized at command (2) to address the next device. This procedure continues until the last entry in the list has been addressed. Command (3) then chains to command (4).

Note 4: Send Ⓓ to terminate addressing.

Note 5: The message is sent to the terminal.

Note 6: The response to longitudinal redundancy check (LRC) is read. Completion (with or without error) is posted in the ECB. If an error is detected, the reset function is not performed. If an error is not detected and the reset option is specified, command (5) chains to command (6).

Note 7: An EOT sequence is sent to set the terminal in control mode.

WRITE Continue (TT) and WRITE Continue with Reset (TTR): A WRITE continue macro instruction is issued after a successful WRITE initial to send (without readdressing) additional message blocks. In addition, WRITE TTR will set the terminal in control mode after the block is sent. The channel program for WRITE TT and TTR is the same as steps 5, 6, and 7 of WRITE TI and WRITE TIR above.

WRITE Positive Acknowledge (TA): This macro instruction is issued by the user to stop receiving message blocks from a terminal before the end of transmission Ⓒ is received. The terminal is reset. The channel program for WRITE positive acknowledge (TA) is a single CCW:

(1) Write ⒹⒸⒸⒸ

The Ⓓ is a positive response and the ⒸⒸⒸ are the EOT sequence.

WRITE Negative Acknowledge (TN): The channel program for WRITE TN is a single CCW:

(1) Write ⒸⒸⒸ

The Ⓒ is interpreted as a negative response. In addition, the line is reset.

The WRITE negative acknowledge macro instruction can also be issued after a WRITE initial (TI) or WRITE continue (TT) to reset the line. The function provided is the same as the TIR or TTR.

Note: The READ and WRITE initial begin with an EOT sequence which resets the line.

Programming Notes

1. To read all of a message, it may be necessary to issue at least two READ macro instructions to poll a terminal component. The first macro instruction must be a READ initial. The second macro instruction must be a READ continue. Figure 22 illustrates channel, line, and 1050 activity for READ initial and READ continue.

Program	Channel	Line	1050
READ initial	1. Write ⒸⒸⒸ	ⒸⒸⒸ →	
	2. Write polling characters	A6	
	3. Read response		← Ⓓ
	4. Read message block		← Block 1 Ⓑ LRC
READ continue	1. Write Ⓨ	Ⓨ →	
	2. Read message block		← Block 2 Ⓑ LRC
READ continue	1. Write Ⓨ	Ⓨ →	
	2. Read message block		← Ⓒ

Figure 22. READ TI and READ TT Macro Instructions

Program	Channel	Line	1050
READ initial	1. Write ⒸⒸⒸ	ⒸⒸⒸ →	
	2. Write polling characters	A6	
	3. Read response		← Ⓓ
	4. Read message block		← Block 1 Ⓑ LRC
WRITE continue	1. Write message block	Ⓓ Block 2 Ⓑ LRC →	
	2. Read response		← Ⓨ

Figure 23. READ TI and WRITE TT Macro Instructions

2. A WRITE continue macro instruction can be issued after any READ, if the last character received is not a Ⓒ. The first character of the message must be a Ⓓ, which selects all ready units of the terminal.

   Note: Arrows to the right specify information from the CPU to the 1050. Arrows to the left specify information from the 1050 to the CPU.

   Figure 23 illustrates channel, line, and 1050 activity for READ initial and WRITE continue.

3. After a READ operation, the user may decide to stop activity of the terminal by issuing:

   a. A WRITE positive acknowledge, if the READ operation was successful;

   b. A WRITE negative acknowledge, if the READ operation was not successful; or

   c. A READ/WRITE initial if the READ operation was not successful, but the user wants to restart activity on the line.

   The 1050 will interpret a Ⓒ character as a negative response.

4. The last macro instruction of a write sequence can be a WRITE initial with reset, a WRITE continue with reset or a WRITE negative acknowledge, if the user wants to reset the line. However, a WRITE initial or a READ initial will reset the line and restart activity.

5. The user must provide the line control characters which pertain to the messages he wishes to send. (See Figure 24.)

No distinction is made in this publication between the lower case and the upper case representation of control characters (i.e., ⓓ and Ⓓ or ⓝ and Ⓝ). Their bit configurations are the same.

MACRO Instruction	First Character	Text	Last Character
WRITE initial	First text character	data	Ⓑ
WRITE initial with reset	First text character	data	Ⓑ
WRITE continue	First text character	data	Ⓑ
WRITE continue with reset	First text character	data	Ⓑ
WRITE continue (conversational mode)	Ⓓ	data	Ⓑ
WRITE continue with reset (conversational mode)	Ⓓ	data	Ⓑ

Figure 24. Line Control Characters for Messages

IBM 1050 TERMINALS ON SWITCHED NETWORKS

Terminal to CPU

Name	Operation	Operand

symbol	DFTRMLST	DIALST, {0 }
		{dialcount-integer,dialchars-decchars}
		({xxyy-hexchars,}...)

Normally, terminal to CPU transmission occurs when:

1. The program issues a READ initial referencing an answering-polling list to enable a line. Prior to the enable, no calls can be received by the transmission control unit.

2. The terminal operator dials the CPU when he has something to send. This completes the enable command, causing a device end-channel end interruption.

3. BTAM polls the terminal that dialed, using the polling characters supplied in the terminal list.

This normal mode of operation is performed using an answering-polling list, defined in the first format shown above:

```
DIALST,0,({xxyy-hexchars,}...)
```

0 Indicates the list is an answering list.

xxyy Hexadecimal representation of the component polling characters as they appear in six-bit BCD transmission code (see Appendix B).

The answering-polling list must contain polling characters that are appropriate for all terminals that can call in on the line. Normally, the common polling character 0 is used to poll all components of a terminal. All terminals should have the same identification character.

Another possible (though not normally appropriate) technique allows terminal to CPU transmission to occur by dialing the terminal and then polling. This technique is useful primarily for unattended operations using card or paper tape input and works as follows:

1. The program issues a READ initial to dial the terminal and then to poll a component.

2. If ready, the component can then begin sending data.

This mode of operation is performed using a call-polling list, defined in the second format shown above:

DIALST, dialcount-integer,
dialchar-decchars, ({xxyy-hexchars,}...)

dialcount Number of digits in the phone number of the terminal.

dialchar Phone number of the terminal.

xxyy Hexadecimal representation of the two polling characters for the component or components.

READ Initial (TI) and READ Initial with Reset (TIR) using Answering-Polling List: The channel program executed for READ initial differs depending on whether the terminal list referred to by the macro instruction is an answering-polling list (normal case) or a call-polling list (exceptional case).

The problem program must issue a READ initial macro instruction to start polling, and the terminal may be reset if desired. The terminal must be connected before data transmission can occur. An answer list is indicated if the first byte of the list is binary zero.

The channel program using an answering-polling list is:

(1)	Disable		Note 1
(2)	Enable		Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	EOT sequence	Note 4
(5)	Write	Polling characters	Note 5
(6)	Read	Response	Note 6
(7)	Read	Message block	Note 7
(8)	Write	ⒹⒸ	Note 8
(9)	Disable		Note 9

Note 1: The disable command resets the line.

Note 2: The enable command conditions the line to receive incoming calls. Before this command is executed, the line is busy to incoming calls. When a call is received, the DE-CE interruption occurs and control returns to BTAM. The channel program is restarted to continue at command (3).

Note 3: Fifteen pad characters are sent.

Note 4: An EOT sequence is sent to place the terminal that dialed into control mode.

Note 5: The polling characters are sent for the next component entry in the polling list.

Note 6: The response is read into the first byte of input area. If the response is negative (single character Ⓝ), the channel program is restarted at command (4) to poll the next component. If the end of an open list is reached before a positive response is received, the operation is posted complete in the ECB. If the response is positive (the first character is Ⓓ) and if the first text character is not Ⓑ or Ⓒ, the channel data chains to (6). If the second character is Ⓑ or Ⓒ, the operation is posted complete in ECB.

Note 7: The message is read until the end of block Ⓑ. The operation is posted complete (with or without error) in the ECB. If READ TIR was specified and an error occurred during transmission, the channel program is terminated. If no error occurred, command (6) chains to command (7).

Note 8: The terminals are placed in control mode.

Note 9: The line connection is broken.

READ Initial (TI) and READ Initial with Reset (TIR) using a Call-Polling List: The channel program using a call-polling list is:

(1)	Disable		Note 1
(2)	Dial	Call Digits	Note 2
(3)	The rest of the channel program is the same as the answering-polling list above.		

Note 1: A Disable command is issued in case the line was not disconnected after the previous operation.

Note 2: The terminal is dialed using the call digits contained in the terminal list.

READ Continue (TT) and READ Continue with Reset (TTR): A READ continue macro instruction is issued following a successful READ TI, TT, TP, or TV to read the next message block from the same component. No polling is performed. The data received will either be the next message block or a Ⓒ that indicates the end of transmission. The reset option may be specified to set the terminal in control mode. The channel program for READ TT and READ TTR is:

(1) Write Positive response Note 1
(2) Rest of channel program is the same as steps 7, 8, and 9 of READ TI.

Note 1: A positive response Ⓨ is sent to the terminal.

<u>READ Repeat (TP) and READ Repeat with Reset (TPR):</u> A READ TP macro instruction is issued following an unsuccessful read operation for retransmission of data that was received in error. The terminal may be reset if desired. The channel program is the same as READ TT, except that it writes a Ⓝ instead of Ⓨ. For paper tape or card input, READ repeat requires the line correction feature. For these components, it can only be used twicé in succession for a total of three tries. Furthermore, it can be used with the 1054 Paper Tape Reader only for message blocks that are equal to or less than 312 characters in length.

<u>READ Conversational (TV) and READ Conversational with Reset (TVR):</u> The READ conversational macro instruction is issued to poll one of the components of a terminal when line connection is already established. The channel program is different from READ TI, in that there is no initial dial or enable. The terminal list referred to by READ conversational must be an open-type polling list. It may be a separate list, defined with the OPENLST operand, or it may be the same list used with READ initial (defined as DIALST). To use the original dial list, the entry operand of the READ TV must refer to "symbol+n+1", where n=number of dial digits.

<u>REAL Skip (TS):</u> The READ skip macro instruction generates a read command with the skip bit on. This macro allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by a user-supplied count. The channel program is a single read skip CCW.

<u>CPU to Terminal</u>

1. The program issues a WRITE initial macro instruction referencing a call-addressing list to dial the terminal and to address the component.

2. If a positive response is received, the message is sent. If there is a negative response, the operation is posted. This normal mode of operation is performed using a dial-addressing list, defined in the second format shown in the DFTRMLST format example:

DIALST,dialcount-integer,dialchar-decchars,({xxyy-hexchars,}...).

dialcount Number of dial digits in the terminal's phone number.

dialchars Phone number of the terminal.

xxyy Two character hexadecimal code as it appears in the 6-bit BCD transmission code (see Appendix B). xx identifies the terminal and must be the same for all entries in the list. yy identifies the component.

Another possible (though normally not appropriate) technique that allows CPU to terminal transmission is:

1. The program issues a WRITE initial referencing an answering-addressing list to enable a line.

2. The terminal operator dials the CPU. This completes the enable command, causing a device end-channel end interruption.

3. BTAM addresses one or more of the components of the terminal that dialed, using the addressing characters supplied in the terminal list. A positive response must be received from each component addressed before the message is sent. This mode of operation is performed using an answering-addressing list, defined in the first format shown in the DFTRMLST format example:

DIALST,0,({xxyy-hexchars,}...).

xxyy Hexadecimal representation of the component's addressing characters as they appear in the six-bit BCD transmission code (see Appendix B).

If this technique is used, the terminal list must be appropriate for all terminals that can call in on the line.

<u>WRITE Initial (TI) and WRITE Initial with Reset (TIR) using Call Addressing List:</u> A WRITE initial macro instruction is issued to send the first block of a message. The channel program for the WRITE initial mácro instruction differs depending on whether the terminal list referred to by the macro instruction is a call-addressing list (normal case) or an answer-addressing list (exceptional case).

For the normal case, using a call-addressing list, the channel program for WRITE TI or WRITE TIR is:

(1)	Disable		Note 1
(2)	Dial	Dial digits	Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	EOT sequence	Note 4
(5)	Write	Addressing characters	Note 5
(6)	Read	Response to addressing	Note 6
(7)	Write	Ⓓ	Note 7
(8)	Write	Message block	Note 8
(9)	Read	Response to VRC/LRC	Note 9
(10)	Write	Ⓒ (TIR only)	Note 10
(11)	Disable	(TIR only)	Note 11

Note 1: A Disable command is issued in case the line was not disconnected after the previous operation.

Note 2: The terminal is dialed.

Note 3: Fifteen pad characters are sent.

Note 4: ⒸⒸⒸ sequence is sent to put terminal in control mode.

Note 5: The addressing characters are written.

Note 6: The response is read into first byte of DECB response field. If the response is positive Ⓨ and the end of list has not been reached, the channel program is restarted at (5) to address the next component. If the response is positive and end of list has been reached, the channel command chains to (7). If the response is negative (for any component in list), post operation complete in ECB.

Note 7: Sends Ⓓ to terminate addressing.

Note 8: Each outgoing message block must begin with a Ⓓ and end with a Ⓑ. BTAM provides the Ⓓ in (7).

Note 9: The response to VRC/LRC check is read. For WRITE initial with reset, if a positive response Ⓨ is received command (7) chains to command (8). Otherwise, the reset function is not performed.

Note 10: The EOT sequence resets the terminal. Line connection is broken.

Note 11: The line is disabled.

WRITE Initial (TI) and WRITE Initial with Reset using Answer-Addressing List: For the exceptional case, using an answer-addressing list, the channel program for WRITE TI or WRITE TIR is:

(1)	Disable	
(2)	Enable	Note 1
(3)	through (11) same as with dial-addressing list.	

Note 1: The enable command is completed when the terminal operator dials the CPU.

WRITE Continue (TT) and WRITE Continue with Reset (TTR): A WRITE continue or WRITE continue with reset macro instruction can be issued following a successful READ TI, TT, TV or TP, or WRITE TI, TT, or TV. The channel program executed for WRITE TT and WRITE TTR is:

(1)	Write	Message block	Note 1
(2)	Read	Response to VRC/LRC	
(3)	Write	Ⓒ (TTR only)	
(4)	Disable	(TTR only)	

Note 1: When the macro is issued after a Write operation, the first character is a text character. After a Read, the first character is a Ⓓ.

No terminal list is used with WRITE TT or WRITE TTR. The remaining commands operate the same as commands 9 through 11 of WRITE TI.

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The channel program for WRITE TV and WRITE TVR is:

(1)	Write	ⒹⒸⒸⒸ	Note 1
(2)	Write	Addressing characters	
(3)	Read	Response to addressing	
(4)	Write	Message block	
(5)	Read	Response to VRC/LRC	
(6)	Write	Ⓒ (TVR only)	
(7)	Disable	(TVR only)	

Note 1: A Ⓓ is sent as a positive response to the message block just read. Commands 2 through 7 operate the same as commands 5, 6, 8, 9, 10, and 11 of WRITE TI.

The terminal list referred to by the WRITE conversational macro instruction must be an open type list. It may be a separate list defined with the OPENLST operand, or it may be the same list used with WRITE initial (defined as DIALST). In order to use the original dial list, the symbol operand of the WRITE TV must refer to "symbol+n+1", where n is equal to the number of dial digits.

WRITE Positive Acknowledge (TA): The problem program issues a WRITE positive acknowledge macro instruction after a successful read operation to stop line activity. A positive response and an EOT sequence is sent. A disable command then breaks the line connection.

The channel program executed for WRITE TA is:

(1) Write ⒹⒸ
(2) Disable

WRITE Negative Acknowledge (TN): The problem program issues a WRITE negative acknowledge macro instruction:

1. After a non-successful read operation to send a negative response and break the line connection.

2. After a write operation to break the line connection.

The channel program executed for WRITE TN is:

(1) Write Ⓒ
(2) Disable

Programming Notes

Line Connection: Line connection on a switched network must be established by a WRITE initial or a READ initial. Connection must be broken by a READ/WRITE with the reset option, a WRITE positive acknowledge, or a WRITE negative acknowledge before another READ/WRITE initial can be issued for this line.

Conversational Mode at Terminal Component Level: A WRITE continue macro instruction can be issued after any READ macro, if the message received is not a Ⓒ. The first character of the message must be a Ⓓ. This option provides the problem program with the facility to answer an inquiry.

Conversational Mode at Terminal Level: The READ/WRITE conversational allows the problem program to initiate further activity on any other component of the same terminal without re-establishing initial contact.

Polling: To poll, the first macro instruction must be either a READ initial (or READ initial with reset) if the line connection is to be established, or a READ conversational (or READ conversational with reset) if the line connection has already been established.

When data is received in error, the line connection is not broken. The user may issue one of the following macro instructions:

1. A READ repeat, to send a negative response and ask for retransmission.

2. A WRITE negative acknowledgment, to reset the line (negative response) and break the line connection.

3. A READ to reset the line (negative response) and restart activity.

4. A READ Conversational with reset will have the same effect as above, but at the end of the operation, the line connection will be broken.

When data is received correctly and the reset option has been specified in the first macro instruction, the line connection is broken. The user must issue a READ initial or WRITE initial macro instruction to restart activity. If the reset option was not specified, the user may issue one of the following macro instructions:

1. A READ continue, to send a positive response and to ask for transmission of the next block of the message.

2. A READ continue with reset will have the same effect as above, but at the end of the operation, the line connection will be broken.

3. A WRITE continue (first character sent is Ⓓ), to send a positive response to the sending device and then a message to all receiving devices of the terminal.

4. A WRITE continue with reset will have the same effect as above, but at the end of the operation the line connection will be broken.

5. A WRITE positive acknowledge, to send a positive response and to break the line connection.

6. A WRITE conversational to send a positive response and address a terminal component.

Addressing: The first macro instruction of an addressing sequence may be:

1. A WRITE initial or a WRITE initial with reset, if the line connection is to be established.

2. A WRITE conversational or a WRITE conversational with reset, if the line connection is already established.

When the terminal receives a message block in error, the problem program may issue a WRITE continue or a WRITE continue with reset macro instruction to retry transmission. However, the first character sent must be a Ⓓ.

IBM 1060 TERMINALS

Terminal to CPU

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST}, {WRAPLST} ({xxyy-hexchars,}...)
symbol	DFTRMLST	{SSALST }, {SSAWLST} {xxyy-hexchars,}...

Each IBM 1062 Teller Terminal is polled with a two-character code. The first of these two polling characters identifies the IBM 1061 Control Unit (normally only one per line, though up to 26 are possible). The second character identifies the 1062 Teller Terminal (one or two per 1061). The four characters xxyy are the hexadecimal representation of the polling characters as they appear in the six-bit BCD transmission code (see Appendix B). Thus, xx identifies the 1061 and yy the 1062. The list may be an open list, a wrap-around list, or an Auto Poll list.

READ Initial (TI) and READ Initial with Reset (TIR) using OPENLST or WRAPLST: A READ initial macro instruction is used to start or restart polling on a line. The channel program for READ TI using OPENLST or WRAPLST is:

(1)	Write	EOT sequence	Note 1
(2)	Write	Polling characters	Note 2
(3)	Read	Response	Note 3
(4)	Read	Message block	Note 4
(5)	Write	ⓎⒸⒸⒸ (TIR only)	Note 5

Note 1: An EOT sequence (ⒸⒸⒸ) is sent to set all terminals on the line in control mode.

Note 2: The polling characters are sent to poll an entry in the list.

Note 3: The response is read into the first two bytes of the input area. If the response is negative (single character Ⓝ), the channel program is re-initialized at command (2) to poll the next entry in the list. If the end of the list is reached before a positive response (Ⓨ and first text character) is received, the operation is posted as complete in the ECB. If the response is positive, command (3) chains to command (4).

Note 4: The message is read until an EOB is received. Then the operation is posted as complete in the ECB.

If READ TIR was specified and no error occurred during transmission, command (4) chains to command (5). Otherwise, the channel program is terminated.

Note 5: Ⓨ and EOT sequence are sent to acknowledge the message and to reset the line.

READ Initial (TI) and READ Initial with Reset (TIR) Using Start-Stop Auto Poll List: The channel program using an open start-stop Auto Poll list (SSALST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting entry	Note 2
(3)	NOP	Note 3
(4)	Read response	Note 4
(5)	Read message	Note 5
(6)	Write ⓎⒸⒸⒸ (TIR only)	Note 6

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which halts the channel program.

Note 3: This is the last command executed when all negative responses are received.

Note 4: This command is executed on a positive response to bring the index byte (corresponding to the responding terminal) and one text byte into the input area.

Note 5: The message is read into the input area until an EOB is received. The operation is posted as complete in the ECB. If READ TIR is specified and no error occurred during transmission, command (5) chains to command (6). Otherwise, the channel program is terminated.

Note 6: On a successful poll Ⓨ is sent to indicate that the read operation occurred without error.

The ⒸⒸⒸ turns the terminal motors off.

READ Initial (TI) and READ Initial with Reset (TIR) using start-stop wraparound Auto Poll list (SSAWLST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	TIC (5)	Note 3
(4)	TIC (7)	Note 4
(5)	Poll first-entry	Note 5
(6)	TIC (5)	Note 6
(7)	Read response	Note 7
(8)	Read text	
(9)	Write ⓎⒸⒸⒸ (TIR only)	

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which restarts polling.

Note 3: On a negative response to polling, at the end of the list, this TIC command is executed to start the second Poll command.

Note 4: On a positive response to polling, this TIC command is executed to start the Read response command.

Note 5: If either Poll command terminates with a negative response at the end of the list, this command restarts polling at the beginning of the list.

Note 6: On a negative response to polling at the end of the list, this TIC command is executed to restart Poll command (5).

Note 7: Commands (7)-(9) operate the same as commands (4)-(6) of READ initial using SSALST.

READ Continue (TT) and READ Continue with Reset (TTR): A READ continue macro instruction is issued after a successful READ TI, TT, or TP to send a positive response and chain to a READ initial sequence.

The channel program for READ TT is:

(1)	Write Response and EOT	Note 1
(2)-(5)	same as (2)-(5) of READ TI.	

Note 1: A positive response character Ⓨ followed by ⒸⒸⒸ is sent over the line.

READ Repeat (TP) and READ Repeat with Reset (TPR): A READ repeat macro instruction is issued after a READ TI, TT, or TP for retransmission of a message block if an error was detected during transmission. The channel program for READ TP is the same as the channel program for READ TT, except that READ TP sends a negative response (Ⓝ) rather than a positive response.

READ Skip (TS): The READ skip macro instruction generates a read command with the skip bit on. This macro allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by a user-supplied count. The channel program for READ TS is a single read skip CCW.

CPU to Terminal

Name	Operation	Operand
symbol	DFTRMLST	OPENLST,(xxyy-hexchars)

Each 1062 on a line is addressed with a two-character code. An addressing terminal list may contain only one terminal address. The first of the two addressing characters identifies the 1061 control unit. The second character identifies the 1062 terminal. The four characters xxyy are the hexadecimal representation of the addressing characters as they appear in the six-bit BCD transmission code (see Appendix B). The list is defined as an open list.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The WRITE initial macro instruction is issued to send a message to a 1062. The channel program executed for WRITE TI and WRITE TIR on 1060 is:

(1)	Write	ⒸⒸⒸ	Note 1
(2)	Write	Addressing characters	Note 2
(3)	Read	Response	Note 3
(4)	Write	Message block	Note 4
(5)	Read	Response	Note 5
(6)	Write	ⒸⒸⒸ (TIR only)	Note 6

Note 1: An EOT sequence (ⒸⒸⒸ) is sent to set all terminals on the line in control mode.

Note 2: The addressing characters are sent, just as they are supplied in the terminal list, to address a terminal.

Note 3: The response to addressing is read into the first byte of the DECB response field. If the response is positive (single character Ⓨ), command (3) chains to command (4). If the response is negative, the operation is posted.

Note 4: Each outgoing message block must begin with Ⓓ and end with Ⓑ. Delete, idle, or other nonprint characters must be included in the output message, where necessary, to allow time for tab, carriage return, and line feed operations.

Note 5: The response to VRC/LRC check is read and the operation is posted complete (with or without error) in the ECB. If the response is positive and WRITE TIR is specified, command (5) chains to command (6). If the response is negative, the chaining does not take place, and the reset function is not performed.

Note 6: An EOT sequence is sent to turn terminal motors off.

WRITE Positive Acknowledge (TA): This macro instruction is issued by the user to send a positive response to the message and to reset the line. The channel program executed for WRITE TA is:

(1) Write ⓎⒸⒸⒸ

WRITE Negative Acknowledge (TN): The problem program issues WRITE TN macro instructions after a non-successful READ operation to send a negative response and to reset the line. WRITE TN is used after a WRITE operation to disconnect the line. The channel program executed for WRITE TN is:

(1) Write ⒸⒸⒸ

AT&T MODEL 33/35 TWX TERMINALS

Terminal to CPU

TWX terminals are not polled. The line connection is established on terminal request; the terminal calls the CPU through a switched network, and the CPU answers the call.

READ Initial (TI) and READ Initial with Reset (TIR): The READ initial (TI) macro instruction is used when transmission is to be originated by the terminal operator dialing the CPU. The terminal list used in this case is an answer list, coded as follows:

Name	Operation	Operand
symbol	DFTRMLST	IDLST,0,numsent-integer, sentchar-hexchars

0 The list is an answer list.

numsent Number of characters sent.

sentchar A string of characters sent on the line when the terminal dials in.

The recommended sequence is: timing character (null), carriage return (CR), line feed (LF), delete, subscriber-selected characters, carriage return, line feed, and Xon. Each of these characters is coded as two hexadecimal digits that represent the character as it appears in the eight-bit data interchange code (see Appendix B).

Example:

Name	Operation	Operand
TWXANSR	DFTRMLST	IDLST,0,14, 01B151FF4B8322A393D313 B15189

These characters represent the following:

01,B1,51,FF = Null, carriage return, line feed, delete

4B,83,22,A3,93,D3,13 = RALEIGH

B1,51,89 = carriage return, line feed, transmitter on.

The CPU identification (i.e., RALEIGH) would be printed at the beginning of a new line, the print element would be positioned to the start of the next line, and the paper tape transmitter would be turned on.

The channel program executed for READ TI and READ TIR for TWX is:

(1)	Disable		Note 1
(2)	Enable		Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	Sequence	Note 4
(5)	Read	Message block	Note 5
(6)	Write	EOT (TIR only)	Note 6
(7)	Disable (TIR only)		

Note 1: A Disable command is issued in

case the line was not previously disabled.

Note 2: The enable command conditions the line to receive incoming calls. Before this command is executed, the line is busy to incoming calls. When a call is received, a device end-channel end interruption occurs and control returns to BTAM. A new EXCP is issued at (3).

Note 3: Fifteen pad characters are sent.

Note 4: The sentchars sequence is sent just as it appears in the terminal list.

Note 5: The message transmitted from the terminal must end with one of the characters: WRU, Xon, Xoff, or the EOT sequence. The EOT sequence breaks the line connection, requiring the next operation to be either another READ initial or a WRITE initial. If the message is terminated by WRU, Xon, or Xoff, the READ conversational (TV) operation can be executed.

Note 6: These are reset functions. The reset functions do not occur if a transmission error is detected. If no error occurs, the terminal is set to control mode and the line connection is broken.

READ Conversational and READ Conversational with Reset: For the READ conversational (TV) macro instruction, the line connection must have previously been established by a READ or WRITE initial. READ TV can also follow a WRITE conversational. The terminal list used in this case is an answer list, coded as follows:

Name	Oper- ation	Operand
symbol	DFTRMLST	IDLST,0,nocnsent-integer, cntrlseq-hexchars

0 The list is an answer list.

nocnsent Number of characters in the control sequence.

cntrlseq Combination of control characters used to prepare the terminal to transmit.

A commonly used sequence is: Xon, 1 to 4 subscriber characters, Xoff.* This may be used provided a WRITE initial was the first sequence. It requires the TD call-in key to be on at the terminal. Xon, alone, may be used, if a READ initial was the first sequence. These sequences start the tape transmitter.

In reading data keyed in by an operator from the TWX keyboard, the problem program may want to send a meaningful sequence of characters to be printed at the TWX. Such a sequence could alert the TWX operator to the fact that the CPU is ready for the next keyboard entry.

The channel program for READ conversational is:

(1)	Write	Control sequence	Note 1
(2)	Read	Data	Note 2
(3)	Write	EOT (TVR only)	
(4)	Disable	(TVR only)	

Note 1: The control sequence prepares the terminal to transmit.

Note 2: Steps (2) through (4) operate the same as steps (5) through (7) of READ TI.

READ Skip (TS): The READ skip macro instruction generates a read command with the skip bit on. This macro allows the user to recover from a lost-data error condition. The rest of the message is read from a terminal to clear the line, but it is not received in main storage. The command reads the amount specified by a user-supplied count. The channel program for READ TS is a single read skip CCW.

CPU to Terminal

TWX terminals are not addressed. Line connection is established on CPU request; the CPU calls the remote terminal through a switched network.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The WRITE initial (TI) macro instruction is used when transmission is to be originated by the CPU calling the terminal. The terminal may be reset after transmission, if desired. The terminal list used in this case is a dial list, coded as follows:

*This sequence is optional and must be wired into the terminal at installation.

Name	Operation	Operand
symbol	DFTRMLST	IDLST,dialcount-integer, dialchars-decchars, numsent-integer, tidseq-hexchars

dialcount Number of dialchars.

dialchars Dial digits of the phone number of the terminal.

numsent Number of characters in the terminal ID sequence.

tidseq String of characters that are to be compared to a terminal-identification sequence sent from the terminal. This ID is sent automatically by the terminal and is determined at the time the terminal is installed.

The channel program for WRITE initial (TI) and WRITE initial with reset (TIR) is as follows:

(1)	Disable		Note 1
(2)	Dial	Dial digits	Note 2
(3)	Read	Terminal-ID-sequence	Note 3
(4)	Write	Data	Note 4
(5)	Write	EOT (TIR only)	Note 5
(6)	Disable	(TIR only)	Note 6

Note 1: The disable command is issued to insure that the line is disconnected before issuing another dial command.

Note 2: The terminal is dialed using the dial digits of the phone number contained in the terminal list.

Note 3: The terminal identification sequence is read. If the sequence sent is equal to the sequence in the terminal list, a new EXCP is issued at command (4). If the sequences are not equal, the operation is posted as complete-with-error.

Note 4: The message should not end with a control character or sequence (e.g. Xon, or Xon, 1-4 characters, Xoff). Command (4) will command chain to command (5) if WRITE TIR is specified and no error is detected during data transmission.

Note 5: An EOT is sent to disconnect the terminal and the channel chains to command (6).

Note 6: For TIR only, the disable command disconnects the line.

Example:

Name	Operation	Operand
RALEIGH	DFTRMLST	IDLST,10,5109280517,17, B1519343B305CDAD83CB4B 05C58DB15189

The terminal ID (tidseq) characters represent the following:

B1,51 - carriage return, line feed

93,43,B3,05,CD,AD,83,CB,4B,
05,C5,8D - IBM 35ASR#1.

B1,51,89 - carriage return, line feed, Xon

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The WRITE conversational (TV) macro instruction can be used following READ initial, READ conversational, WRITE initial, or a previous WRITE conversational. The channel program for WRITE TV and WRITE TVR for TWX is:

(1)	Write	Data	Note 1
(2)	Write	EOT (TVR only)	
(3)	Disable	(TVR only)	

Note 1: The data transmitted must not end with a control character. The reset function does not occur if a transmission error is detected. The steps are the same as 4, 5, and 6 of WRITE TIR.

WRITE Negative Acknowledge (TN): The WRITE negative acknowledge (TN) macro instruction is used to turn off the terminal motors and disconnect the line. It may be used to disconnect the line after any READ or WRITE operation issued without reset. The channel program executed for WRITE TN on TWX is:

(1)	Write	EOT
(2)	Disable	

IBM 2740 COMMUNICATIONS TERMINAL

The IBM 2740 Communications Terminal may be tailored for the user's application. BTAM includes support for the basic 2740 and the checking, dial, transmit control, and station control features that are available on the 2740. The different features are referred to as follows: "C" for checking, "D" for dial, "T" for transmit control, and "S" station control.

Two restrictions exist in installing the 2740 features: station control and dial are exclusive, and dial is a prerequisite for transmit control.

The following eight types of 2740 are supported by BTAM:

1. 2740 (basic)
2. 2740C (with checking; may have 2760 Optical Image Unit added)
3. 2740D (with dial)
4. 2740DC (with dial and checking; may have 2760 Optical Image Unit added)
5. 2740DT (with dial and transmit control)
6. 2740DTC (with dial, transmit control, and checking)
7. 2740S (with station control)
8. 2740SC (with station control and checking)

The checking feature is the only one that informs the problem program that a message has been received. Without the checking feature, a READ macro instruction may be associated with a WRITE macro instruction. The receipt of a message by a 2740 without checking is indicated by the terminal operator who sends either a positive or negative response.

The Prepare command terminates when the first character is received, but the character is not stored in main storage. This character is normally an End of Addressing character Ⓓ which is generated when the user depresses the BID key and sets the line to control mode. Therefore, when the problem program issues a READ macro instruction which results in a Prepare-Read CCW sequence, the Ⓓ should be restored by the problem program before an attempt is made to write a message.

2740 BASIC

Terminal to CPU

Point-to-point operations are supported for the 2740 basic using a dedicated line. These operations are performed on a contention basis with a terminal seizing the line by depressing the BID key. The BID key sends a Ⓓ to indicate to the CPU that the terminal has data to be read.

The DFTRMLST macro instruction is not used because there is no polling of the 2740 basic.

READ Initial (TI) and READ Initial with Leading Reset (TIR): The READ initial macro instruction performs a read operation. The channel program for READ TI on the 2740 basic is:

(1) NOP
(2) Prepare
(3) Read Data

The NOP is changed to a WRITE EOT when the reset option is specified.

The prepare command conditions the control unit to receive a message, and chains to the read data command when a character is received. The Ⓓ , sent by the transmitting terminal upon depression of the BID key, is deleted by the prepare command. The text, received by the CPU and all terminals on the line, ends with a Ⓒ which causes the 2740 to revert to standby mode. Another terminal can then seize the line by depressing the BID key and sending a Ⓓ .

READ Skip (TS): The READ skip macro instruction generates a read command with the skip bit on which allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by a user-supplied count. The channel program is a single read skip CCW.

CPU to Terminal

The DFTRMLST macro instruction does not apply because there is no addressing of the 2740 basic.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The WRITE initial or WRITE initial with reset macro instruction is issued to send a message to a terminal. In addition, the WRITE TIR deselects the terminal. The channel program is:

(1)	Write	Characters	Note 1
(2)	Write	Data	Note 2
(3)	Write	Ⓒ Ⓒ Ⓒ (TIR only)	Note 3

Note 1: A Ⓓ is sent followed by fifteen idle characters.

Note 2: The text is sent to the terminal. If the reset option was specified, the channel chains to command (3).

Note 3: The three Ⓒ's are sent to place the terminal in standby mode.

WRITE Continue (TT) and WRITE Continue with Reset: The WRITE continue or WRITE continue with reset is issued after a successful WRITE Initial to send additional message blocks. In addition, WRITE TTR sets the terminal in control mode after the message block is sent. The channel program for WRITE TT and WRITE TTR is identical to commands 2 and 3 of WRITE TI and WRITE TIR above.

Programming Note: The WRITE TT and WRITE TTR can only be used after the terminal has been placed in text mode by a WRITE TI. Successive uses of WRITE TI after the terminal has been placed in text mode by the first WRITE TI cause the Ⓓ to be printed as a pound sign (#).

The Read TIR provides a means of resetting the 2740 before allowing the remote operator to enter a message. This permits the problem program to leave the 2740 in text mode after a WRITE operation without the reset option until the problem program is ready to receive a message. The problem program can then issue a READ TIR to return the terminal to control mode. This prevents any characters from being entered before the program is ready to receive them.

2740 WITH CHECKING

The 2740 communications terminal with the checking feature allows point-to-point operations between a terminal and the CPU. The DFTRMLST macro does not apply because there is no polling or addressing of the 2740 with checking. The 2760 Optical Image Unit may be attached to this model of the 2740.

Terminal to CPU

The line is seized by depressing the BID key of the Selectric typewriter or by pressing the light pen against a response of the 2760 Optical Image Unit, which sends a Ⓓ over the line. The message is sent and is followed by a Ⓑ .

READ Initial (TI) and READ Initial with Reset (TIR): The READ initial or the READ initial with reset macro instruction is issued to receive a message from a terminal. The channel program for READ TI and READ TIR is:

```
(1)  NOP
(2)  Prepare
(3)  Read   Data
(4)  Write  Ⓓ Ⓒ Ⓒ Ⓒ  (TIR only)
```

The NOP is changed to a WRITE EOT when the reset option is specified.

The prepare command conditions the control unit to receive a message and then chains to the read data command when a character is received. the Ⓓ sent by the transmitting terminal (upon depression of the BID key of the Selectric typewriter or by pressing the light pen against a response point of the 2760 Optical Image Unit) is deleted by the prepare command. The data (text) will end with a Ⓑ. If the reset option is specified and no error occurred during transmission, the CPU sends a Ⓓ to indicate the message was received without error and sends Ⓒ Ⓒ Ⓒ to turn the terminal motors off and place the line in control mode.

READ Continue (TT) and READ Continue with Reset (TTR): These macro instructions are issued after a successful read operation to read another message from the same terminal. The channel program for READ TT or READ TTR is:

```
(1)  Write  Ⓨ
(2)  Read   Data
(3)  Write  Ⓓ  Ⓒ  Ⓒ  Ⓒ  (TIR only)
```

The Ⓨ is a positive response sent by the CPU to the terminal to indicate the previous message was received successfully. Command chaining to the read takes place to read the next message. The first character of the message is either a text character or an EOT. If the first character is a text character, the message is read. An EOT ends transmission. If the reset option was specified and no errors occurred during transmission, the Ⓓ (positive response to message) and three Ⓒ's are sent to turn the terminal motors off.

Note: The READ Continue operations (READ TT and READ TTR) are not supported by BTAM for the 2760 Optical Image Unit attached to a 2740 with the checking features.

READ Repeat (TP) and READ Repeat with Reset (TPR): The READ repeat and READ repeat with reset macro instructions are issued for retransmission of a message which previously was received in error. The channel program for READ TP and READ TPR is:

(1)	Write	Ⓝ
(2)	Read	Data
(3)	Write	Ⓓ Ⓒ Ⓒ Ⓒ (TPR only)

The negative response Ⓝ is sent to indicate an error occurred during transmission of the previous message. Command chaining to the read data command takes place. The previous message is read again. If the reset option was specified and no errors occurred during transmission, the reset sends a positive acknowledgment Ⓓ and turns the terminal motors off with Ⓒ Ⓒ Ⓒ.

<u>READ Skip (TS):</u> The READ skip macro instruction allows the user to recover from lost-data error conditions. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The channel program for READ TS on 2740 with checking is:

(1)	Read	Data

The read data command reads the amount specified by a user-supplied count. The skip bit is turned on in the CCW flags.

<u>CPU to Terminal</u>

The CPU sends a Ⓓ to put the terminal in receive mode. The message is then sent by the CPU and is followed by a Ⓑ.

<u>WRITE Initial (TI) and WRITE Initial with Reset (TIR):</u> The WRITE initial and WRITE initial with reset macro instructions are issued to send a message to the terminal and to read a response to the LRC check. If reset is specified the terminal motors are turned off. The channel program for WRITE TI and WRITE TIR is:

(1)	Write	Characters
(2)	Write	Data
(3)	Read	Response
(4)	Write	Ⓒ Ⓒ Ⓒ (TIR only)

The CPU sends the Ⓓ to put the terminal(s) in receive mode and also writes fifteen idle characters to give the terminal motors time to get the necessary speed. The message is sent and then the response (positive Ⓨ or negative Ⓝ) to VRC/LRC check is read. If the reset option was specified and no errors occurred, the terminal motors will be turned off by the Ⓒ Ⓒ Ⓒ.

<u>WRITE Initial Optical (TIO)</u> macro transmits a control message to the 2760 Optical Image Unit and reads its response. If the response is satisfactory, a Ⓒ is sent to the 2760 OIU to enable it to execute the function specified in the control message. If the response is unsatisfactory, Ⓒ is not sent and the error is indicated to the user. The channel program for WRITE TIO is:

(1)	Write	Ⓓ (PRE)(O)	Note 1
(2)	Write	Frame Change	Note 2
(3)	Write	Ⓑ	Note 1
(4)	Read	VRC/LRC Response	
(5)	Write	ⒸⒸⒸ	Note 3

Note 1: These characters are supplied by BTAM and should therefore not be coded by the user. The (PRE)(O) represent the prefix character and letter O, and they select the 2760 Optical Image Unit.

Note 2: Three frame change characters in the form of (F)(A$_1$)(A$_2$), where (F) describes direction of film movement and end of message mode, and (A$_1$)(A$_2$) describe the amount of film movement. The address of these characters should be placed in the entry operand of the WRITE macro. The length operand will be ignored.

Note 3: The reset option is not allowed for WRITE TIO, since successful completion leaves the line in control mode. If the macro is completed in error, the user should issue a WRITE TN macro to clear the error condition before retrying the WRITE TIO.

<u>WRITE Continue (TT) and WRITE Continue with Reset (TTR):</u> The WRITE continue and WRITE continue with reset macro instructions are issued after a successful WRITE operation to send a message block. The same message or another message may be written depending on the area address specified in the write data command. The Ⓓ is not sent before the message. The channel program for WRITE TT and WRITE TTR is:

(1)	Write	Data
(2)	Read	Response
(3)	Write	Ⓒ Ⓒ Ⓒ (TTR only)

The message is sent and ends with Ⓑ. The terminal will send a positive response Ⓨ

or a negative response Ⓝ. If the reset option was specified and no errors occurred, the terminal motors will be turned off.

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The WRITE conversational and the WRITE conversational with reset macro instructions are issued after a successful read operation or for reestablishing connection with the Selectric typewriter after using the Optical Image Unit. The channel program for WRITE TV and WRITE TVR is:

(1) Write Ⓓ
(2) Write Data
(3) Read Response
(4) Write Ⓒ Ⓒ Ⓒ (TVR only)

WRITE TV or TVR sends a Ⓓ after a read operation to turn the terminal around, putting it in receive mode. After the data which ends with a Ⓑ is sent, a response either positive or negative is read. If the reset option was specified and no errors have occurred during transmission, the Ⓒ Ⓒ Ⓒ will turn the terminal motors off.

WRITE Invitational Optical (TCO) macro transmits a Ⓓ and a control message to the 2760 Optical Image Unit and reads its response. If the response is satisfactory, a Ⓒ is sent to the 2760 OIU to enable it to execute the function specified in the control message, and the 2760 OIU is read immediately. If the response is unsatisfactory, Ⓒ is not sent and the error is indicated to the user. The channel program for WRITE TCO is:

(1) Write Ⓓ (PRE)(O) Note 1
(2) Write Frame Change Note 2
(3) Write Ⓑ Note 1
(4) Read VRC/LRC Response
(5) Write ⒸⒸⒸ
(6) Prepare
(7) Read Data Note 3

Note 1: These characters are supplied by BTAM and should therefore not be coded by the user. The (PRE)(O) represent the prefix character and letter O, and they select the 2760 Optical Image Unit.

Note 2: Three frame change characters in the form of $(F)(A_1)(A_2)$, where (F) describes direction of film movement and end of message mode, and $(A_1)(A_2)$ describe the amount of film movement. The address of these characters should be placed in the entry operand of the WRITE macro. The length operand will be ignored.

Note 3: The address for the input area should be coded in the area operand and the length in the length operand of the WRITE macro. The length may be 7 bytes or variable, depending on the end of message mode.

WRITE Positive Acknowledge (TA): The WRITE positive acknowledge macro instruction is sent after a successful read operation to send a positive response and to turn the terminal motors off. The channel program for WRITE TA on the 2740 with checking is a single CCW which writes a Ⓓ and Ⓒ Ⓒ Ⓒ.

WRITE Negative Acknowledge (TN): The WRITE negative acknowledge macro instruction is issued after a read or write operation. Issued after a read operation, WRITE TN is a negative response which turns the terminal motors off. If issued after a write operation, it turns the terminal motors off. The channel program for WRITE TN is a single CCW that writes Ⓒ Ⓒ Ⓒ.

2740 WITH DIAL

With the dial-up adapter feature installed, the 2740 can communicate over a common-carrier dial network.

Terminal to CPU

Each terminal that has the dial feature must first establish the line connection by calling the CPU. After this is done, the terminal's BID key can be depressed, thus sending the Ⓓ.

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0

The 0 specifies this is an answer list.

READ Initial (TI) and READ Initial with Reset (TIR): A READ initial macro instruction is issued to perform a read operation. The READ initial with reset macro instruction also reads and then resets the line. The channel program for READ TI and READ TIR is:

(1)	Disable		Note 1
(2)	Enable		Note 2
(3)	Read	Message	Note 3
(4)	Write	Ⓒ (TIR only)	Note 4
(5)	Disable	(TIR only)	Note 5

Note 1: The disable command will reset the line in case it was not disabled before.

Note 2: The line is enabled to receive a call. When a call is received, this command chains to (3).

Note 3: The message is read into main storage. The message format is a Ⓓ and text. The operation is posted complete (with or without error) in the ECB. If the reset option was specified and no errors have occurred during transmission, this command chains to (4). Otherwise, there is no chaining.

Note 4: The Ⓒ turns the terminal motors off.

Note 5: The line connection must be dropped when transmission is complete. This may be done by the CPU with the disable command that is part of the reset function.

<u>READ Conversational (TV) and READ Conversational with Reset (TVR):</u> The READ TV and READ TVR macro instructions are issued after a successful write operation to perform a read operation. In addition the TVR will reset the line by turning the terminal motors off and then disabling the line. The channel program for READ TV and READ TVR is:

(1)	Prepare		Note 1
(2)	Read	Data	Note 2
(3)	Write	Ⓒ (TVR only)	Note 3
(4)	Disable	(TVR only)	Note 4

Note 1: The prepare command conditions the control unit to receive a message and then chains to the read data command when a character is received.

Note 2: The text is read.

Note 3: The Ⓒ is sent to turn the terminal motors off.

Note 4: The line is disabled.

<u>READ Skip (TS):</u> The READ skip macro instruction is issued to allow the user to recover from lost-data error conditions. The rest of the message is read from the terminal to clear the line, but it is not received into main storage. The macro generates a single CCW which is a read command with the skip bit on. The command reads the amount specified by a user-supplied amount.

<u>CPU to Terminal</u>

The CPU may call the terminal so that the terminal may receive a message.

<u>WRITE Initial (TI) and WRITE Initial with Reset (TIR):</u> The WRITE initial and the WRITE initial with reset macro instructions are issued to send a message to the terminal. The CPU dials the terminal. The DFTRMLST is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars

The dialcount is the number of digits in the phone number of the terminal and dialchar is the phone number of the terminal. The channel program for WRITE TI and WRITE TIR on the 2740 with dial is:

(1)	Disable		Note 1
(2)	Dial	Call digits	Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	Message	Note 4
(5)	Write	Ⓒ (TIR only)	Note 5
(6)	Disable	(TIR only)	

Note 1: The disable command disables the line in case it was not done before.

Note 2: The CPU then calls the terminal using the dial digits specified in the DFTRMLST.

Note 3: Fifteen pad characters are sent.

Note 4: The message is sent to the terminal. The operation is posted complete (with or without error) in the ECB. If the reset option is specified and no errors have occurred during transmission, the reset function is performed.

Note 5: The Ⓒ is sent to turn the terminal motors off and the line is disconnected by the disable command.

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): These macro instructions are issued after a successful read operation to send a message to the terminal with which connection is already established. The channel program for WRITE TV and WRITE TVR is:

```
(1)  Write    Data
(2)  Write    Ⓒ (TVR only)
(3)  Disable (TVR only)
```

WRITE Disconnect (TN): The WRITE disconnect macro instruction is issued to disconnect the line after a read or write operation. The channel program for WRITE TN is:

```
(1)  Write    Ⓒ
(2)  Disable
```

2740 WITH DIAL AND CHECKING

The 2740 with the dial and checking features requires that each terminal have the dial-up adapter special feature installed. The line connection is established by either the terminal calling the CPU or the CPU calling the terminal. The 2760 Optical Image Unit may be attached to this model of 2740.

Terminal to CPU

The DFTRMLST for terminal-to-CPU operations is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0

The 0 specifies that this is an answer list.

READ Initial (TI) and READ Initial with Reset (TIR): The READ initial and the READ initial with reset macro instructions are issued to perform a read after the terminal has dialed the CPU. The channel program for READ TI and READ TIR is:

```
(1)  Disable                        Note 1
(2)  Enable                         Note 2
(3)  Read     Data                  Note 3
(4)  Write    Ⓓ  Ⓒ (TIR only)       Note 4
(5)  Disable (TIR only)             Note 5
```

Note 1: The line is disabled in case it was not disabled previously.

Note 2: The line is enabled to receive a call from the terminal. When a call is received, the enable command chains to (3).

Note 3: The message is read into main storage. The message will begin with a Ⓓ caused by the Selectric typewriter's BID key being depressed or by pressing the light pen against a response point of the 2760 Optical Image Unit. The message will end with a Ⓑ which is sent when the terminal operator depresses the EOB key, or automatically by the 2760 Optical Image Unit when in Auto EOM mode. The operation is posted complete with or without error in the ECB. If the reset option was specified and no errors occurred during transmission, the reset function is performed.

Note 4: The Ⓓ is a positive response indicating to the terminal that the data was received successfully. The Ⓒ will turn the terminal motors off.

Note 5: The line is disabled.

READ Continue (TT) and READ Continue with Reset (TTR): The READ continue and the READ continue with reset macro instructions are issued after a successful read operation to send a positive response and to read more data.

Note: The READ Continue operations (READ TT and READ TTR) are not supported by BTAM for the 2760 Optical Image Unit attached to a 2740 with both the dial and checking features.

The channel program for READ TT and READ TTR is:

```
(1)  Write    Ⓨ                     Note 1
(2)  Read     Data                  Note 2
(3)  Write    Ⓓ  Ⓒ (TTR only)       Note 3
(4)  Disable (TTR only)             Note 4
```

Note 1: A positive response Ⓨ to the previous read is sent and this command chains to (2). The line connection has previously been established.

Note 2: The data is read into main storage. Since connection was previously established, a Ⓓ does not precede the data. The text ends with a Ⓑ.

Note 3: If there were no errors and reset was specified, the Ⓓ is sent as a positive response and the Ⓒ turns the terminal motors off.

Note 4: The line is disabled.

READ Conversational (TV) and READ Conversational with Reset (TVR): The READ TV and READ TVR macro instructions are issued after a successful write operation, or for reestablishing connection with the Selectric typewriter after using the 2760 Optical Image Unit, to read data from the terminal. In addition, the READ TVR will reset the line. The channel program for READ TV and READ TVR is:

(1) Write Ⓒ Ⓒ Ⓒ Note 1
(2) Prepare Note 2
(3) Read Data Note 3
(4) Write Ⓓ Ⓒ (TVR only)
(5) Disable (TVR only)

Note 1: Three Ⓒ's are sent to put the terminal in control mode.

Note 2: The prepare command conditions the control unit to receive a message and chains to the read data command when a character is sent.

Note 3: The D sent by the transmitting terminal (upon depression of the BID key or by pressing the light pen against a response point of the 2760 Optical Image Unit) is deleted by the prepare command. The text follows until a Ⓑ is received.

READ Repeat (TP) and READ Repeat With Reset (TPR): A READ repeat or a READ repeat with reset macro instruction is issued to retransmit a message received in error. The channel program for READ TP or READ TPR on the 2740 with dial and checking is:

(1) Write Ⓝ
(2) Read Data
(3) Write Ⓓ Ⓒ (TPR only)
(4) Disable (TPR only)

READ Skip (TS): A READ skip macro instruction generates a read command with the skip bit on. This macro allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by the user-supplied count. The channel program for READ TS is a single read skip CCW.

CPU to Terminal

CPU-to-terminal operations require that the CPU dial the terminal. The DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer dialchar-decchars [,(ffaaaa-hexchars)]

The dialcount represents the number of digits in the telephone number of the terminal. The dialchar is the actual phone number of the terminal. "ffaaaa" represents (F)(A_1)(A_2), the three frame change characters sent to the Optical Image Unit attachment to describe film movement.

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The WRITE initial or WRITE initial with reset macro instruction is issued to dial the terminal and to send it a message. The channel program for WRITE TI or WRITE TIR is:

(1) Disable Note 1
(2) Dial Dial digits Note 2
(3) Write Pad characters Note 3
(4) Write Data Note 4
(5) Read Response Note 5
(6) Write Ⓒ (TIR only) Note 6
(7) Disable (TIR only) Note 7

Note 1: The line is disabled in case it was not disabled previously.

Note 2: The CPU dials the terminal, using the dial digits specified in the DFTRMLST.

Note 3: Fifteen pad characters are sent after the terminal is dialed.

Note 4: The data is sent to the terminal. The text begins with a Ⓓ and ends with a Ⓑ.

Note 5: The response is read to determine if the transmission occurred without error.

Note 6: If the reset option was specified and no errors occurred, a Ⓒ is sent to turn the terminal motors off.

Note 7: The line is disabled.

WRITE Initial Optical (TIO): macro makes initial contact with the terminal, transmits a control message to the 2760 OIU, and reads its response. If the response is satisfactory, a Ⓒ is sent to the 2760 OIU to enable it to execute the function specified in the control message. If the response is unsatisfactory, Ⓒ is not sent and the error is indicated to the user. The channel program for WRITE TIO is:

(1)	Disable		
(2)	Dial	Digits	
(3)	Write	Pad Characters	
(4)	Write	Ⓓ (PRE)(O)	Note 1
(5)	Write	Frame Change	Note 2
(6)	Write	Ⓑ	Note 1
(7)	Read	VRC/LRC Response	
(8)	Write	Ⓒ Ⓒ Ⓒ	Note 3

Note 1: These characters are supplied by BTAM and should not be coded by the user. The (PRE)(O) represent the prefix character and letter O, and they are used to select the Optical Image Unit.

Note 2: The user must prepare a DFTRMLST DIALST to supply the three frame change characters following the dial characters. The frame characters are in the form of (F)(A-1)(A-2), where (F) describes the direction of film movement and end of message mode, and (A-1)(A-2) describe the amount of film movement. The length operand will be ignored.

Note 3: The reset option is not allowed for WRITE TIO, since successful completion leaves the line in control mode.

WRITE Continue (TT) or WRITE Continue with Reset (TTR): The WRITE continue or the WRITE continue with reset macro instruction is sent after a write operation. It may be used, if the previous write operation was successful, to send another message block. If an error occurred on a previous write operation, the WRITE TT or TTR can write the same message to the terminal. The channel program for WRITE TT or WRITE TTR is:

(1)	Write	Data
(2)	Read	Response
(3)	Write	Ⓒ (TTR only)
(4)	Disable	(TTR only)

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): These macro instructions are issued after each successful read operation, or for reestablishing connection with the Selectric typewriter after using the 2760 Optical Image Unit, to send a message to the terminal just read. The channel program is:

(1)	Write	Ⓓ (Positive response to previous message)
(2)	Write	data
(3)	Read	response
(4)	Write	Ⓒ (TVR only)
(5)	Disable	(TVR only)

WRITE Invitational Optical (TCO) macro transmits a Ⓓ and a control message to the 2760 OIU and reads its response. If the response is satisfactory, a Ⓒ is sent to the 2760 OIU to enable it to execute the function specified in the control message, and the OIU is read immediately. If the response is unsatisfactory, Ⓒ is not sent and the error is indicated to the user. The channel program for WRITE TCO is:

(1)	Write	Ⓓ (PRE)(O)	Note 1
(2)	Write	Frame Change	Note 2
(3)	Write	Ⓑ	Note 1
(4)	Read	VRC/LRC Response	
(5)	Write	Ⓒ Ⓒ Ⓒ	
(6)	Prepare		
(7)	Read	Data	Note 3

Note 1: These characters are supplied by BTAM and should therefore not be coded by the user. The (PRE)(O) represent the prefix character and letter O, and they select the 2760 Optical Image Unit.

Note 2: Three frame change characters in the form of (F)$(A_1)(A_2)$, where (F) describes the direction of film movement and end of message mode, and $(A_1)(A_2)$ describe the amount of film movement. The address of these characters should be placed in the entry operand of the WRITE macro. The length operand will be ignored.

Note 3: The address for the input area should be coded in the area operand and the length in the length operand of the WRITE macro. The length may be 7 bytes or variable, depending on the end of message mode.

WRITE Positive Acknowledge and Disconnect (TA): The WRITE TA macro instruction is issued after a successful read operation to send a positive acknowledgment to the terminal, to turn the terminal motors off, and to disconnect the line. The channel program for WRITE TA on the 2740 with dial and checking is:

(1) Write Ⓓ Ⓒ
(2) Disable

<u>WRITE Negative Acknowledge and Disconnect (TN):</u> The WRITE TN macro instruction can be issued after a read operation to indicate a negative response to the previous read, to turn the terminal motors off, and to disable the line. It can be issued after a write operation to disconnect the line. The channel program for WRITE TN is:

(1) Write Ⓒ
(2) Disable

2740 WITH DIAL AND TRANSMIT CONTROL

The 2740 with dial and transmit control permits the CPU to control the send and receive status of the terminal. The transmit control feature can be installed only when the dial-up feature is also installed. Provided with the transmit control feature is the transmit control switch located on the left side of the 2740 cabinet. This switch (MTC, OFF) permits the terminal operator to set up the terminal for proper line control when communicating with the CPU. The switch must be set to MTC when using the transmit control code (/and space).

Terminal to CPU

The CPU can control the status of a remote terminal to send information with the two-character control code. When the /space code is sent, the terminal transmit light comes on, <u>if</u> the operator has the transmit control switch on MTC. If the terminal does not send data within 15 seconds, the CPU times-out and the standby light comes on. The terminal automatically sends a Ⓓ in response to the /space but data must follow or the time-out will occur. The DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, {0 {dialcount-integer, dialchar-decchars}

<u>READ Initial (TI) and READ Initial with Reset (TIR):</u> The READ initial and the READ initial with reset macro instructions are issued to perform a read operation. When the READ TI is pointing to a call list, the format of the DFTRMLST operand field used is:

DIALST,dialcount-integer,
dialchar-decchars

The dialcount is the number of dial digits in the terminal's phone number. The dialchars is the phone number of the terminal. The channel program for READ TI or READ TIR with a dial list is:

(1)	Disable		Note 1
(2)	Dial	Digits	Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	Selection characters	Note 4
(5)	Read	Response	Note 5
(6)	Read	Data	Note 6
(7)	Write	Ⓒ (TIR only)	Note 7
(8)	Disable	(TIR only)	

Note 1: The line is disabled in case it was not disabled previously.

Note 2: The CPU calls the terminal using the dial digits specified in the DFTRMLST.

Note 3: Fifteen pad characters are sent to the terminal.

Note 4: The two-character code (/ and space) is sent to allow the terminal to send a message.

Note 5: Read response to see if the terminal has anything to send. If nothing is sent within 15 seconds a time-out occurred. If data is received, this command chains to (6).

Note 6: The rest of the message is read into main storage. If the reset option was specified and no errors occurred during transmission, this command will chain to (7). The operation is posted complete with or without error in the ECB.

Note 7: A Ⓒ is sent, which turns the terminal motors off, and the line is disabled.

The CPU may also answer a call from a terminal.

This gives the terminal the ability to call when it has data to send instead of waiting to be polled. The format of the DFTRMLST operand field is:

DIALST,0

The 0 specifies that this is an answer list. When the READ TI and READ TIR points to an answer list, the channel program is:

(1)	Disable		Note 1
(2)	Enable		Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	Selection characters	Note 4
(5)	Read	Response	Note 5
(6)	Read	Data	
(7)	Write	Ⓒ (TIR only)	
(8)	Disable	(TIR only)	

Note 1: The line is disabled in case it was not disabled previously.

Note 2: The line is enabled until a call is received. The enable command is terminated.

Note 3: Fifteen pad characters are sent.

Note 4: The two-character code (/ and space) is sent by the CPU to indicate to the terminal that it can begin sending data.

Note 5: The CPU reads a response (Ⓓ followed by data) and command chains to (6) if data is received. If data is not received within 15 seconds, there is a time-out.

READ Conversational (TV) and READ Conversational With Reset (TVR): The READ TV and READ TVR macro instructions are issued to read without re-establishing initial contact. They are issued after a successful read operation that did not reset the line. The channel program for READ TV and READ TVR on the 2740 with dial and transmit control is:

(1)	Write	Selection characters	Note 1
(2)	Read	Response	
(3)	Read	Data	
(4)	Write	Ⓒ (TVR only)	
(5)	Disable	(TVR only)	

Note 1: The two-character selection code (/ and space) is sent to the terminal to indicate that it can now send data to the CPU.

READ Skip (TS): The READ skip macro instruction allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal but it is not received in main storage. The channel program is a single CCW that is a read command with a skip flag on. The command reads the amount specified by a user-supplied count.

CPU to Terminal

The CPU dials the terminal to send a message. The DFTRMSLT used is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, {0 {dialcount-integer, dialchar-decchars}

WRITE Initial (TI) and WRITE Initial with Reset (TIR): The WRITE TI and WRITE TIR macro instructions are issued to send data to a remote terminal. When the CPU is to call the terminal, the operand field of the DFTRMLST is:

DIALST, dialcount-integer,
dialchar-decchars

The dialcount is the number of digits in the phone number of the terminal and the dialchar is the actual phone number of the terminal. The channel program for WRITE TI and WRITE TIR when they point to a dial list on 2740 with dial and transmit control is:

(1)	Disable	
(2)	Enable	
(3)	Write	15 pad characters
(4)	Write	Data
(5)	Write	Ⓒ (TIR only)
(6)	Disable	(TIR only)

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The WRITE TV and WRITE TVR macro instructions are issued after a successful read operation to send data in answer to the data read. The channel program for WRITE TV and WRITE TVR is:

(1)	Write	Data
(2)	Write	Ⓒ (TVR only)
(3)	Disable	(TVR only)

WRITE Disconnect (TN): The WRITE disconnect macro instruction is issued to reset the line. The channel program for WRITE TN on 2740 with dial and transmit control is:

(1)	Write	Ⓒ
(2)	Disable	

2740 WITH DIAL, TRANSMIT CONTROL, AND CHECKING

The line connection is established by dialing and is maintained until the CPU issues a disable command. The line connection can also be broken by depression of the dial-disconnect key at the terminal.

The transmit control feature allows the CPU to control the sending and receiving status of the terminal. A two-character control code (/ and space) is sent to indicate the terminal can begin sending data. The transmit control feature can be installed only when the dial feature is installed. The transmit control switch must be set to MTC when using the two-character code.

The checking feature consists of VRC (vertical redundancy check) which checks each transmitted character for parity, and LRC (longitudinal redundancy check) which checks each message block for correct transmission.

Terminal to CPU

The CPU controls the terminal sending data with the two-character code (/ and space). Either the CPU or the terminal may dial the other to establish line connection.

READ Initial (TI) and READ Initial with Reset (TIR): The READ TI and READ TIR macro instructions are issued to perform a read operation. When the terminal establishing the line connection dials the CPU, the format of the DFTRMLST is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0

The 0 specifies this is an answer list. The channel program for READ TI or READ TIR when the terminal dials the CPU is:

(1)	Disable		Note 1
(2)	Enable		Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	Selection characters	Note 4
(5)	Read	Response	Note 5
(6)	Read	Data	Note 6
(7)	Write	Ⓓ Ⓒ (TIR only)	Note 7
(8)	Disable	(TIR only)	Note 8

Note 1: The line is disabled in case it was not done previously.

Note 2: The enable command conditions the line so that the terminal may call the CPU. This command is chained to (3) when a terminal calls in.

Note 3: Fifteen pad characters are sent.

Note 4: The two-character selection code (/ and space) is sent to allow the terminal to begin sending.

Note 5: The response is read. If an EOT is received, the operation is posted complete. If data is received, the command chains to (6) to read the message block.

Note 6: The rest of the message is read. The operation is posted complete (with or without error) in the ECB. If the reset option was specified and no errors occurred during transmission, this command chains to (7). Otherwise, the reset option is not performed.

Note 7: The Ⓓ is sent to indicate good transmission and the Ⓒ is sent to turn the terminal motors off.

Note 8: The line is disabled.

If the CPU dials the terminal and establishes line connection, the format of DFTRMLST is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars

The dialcount is the number of dial digits in the terminal's phone number. The dialchar is the phone number of the terminal. The channel program for READ TI or READ TIR when pointing to a call list is:

(1)	Disable		Note 1
(2)	Dial		Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	Selection characters	Note 4
(5)	Read	Response	Note 5
(6)	Read	Data	Note 6
(7)	Write	Ⓓ Ⓒ (TIR only)	
(8)	Disable	(TIR only)	

Note 1: The line is disabled in case it was not disabled previously.

Note 2: The CPU dials the terminal using the number specified in the DFTRMLST.

Note 3: Fifteen pad characters are sent.

Note 4: The two-character selection code (/ and space) are sent to the terminal to allow it to begin sending data.

Note 5: A two-byte response is read. If an EOT is received, the operation is posted complete. If data is received, the command chains to (6) to continue reading.

Note 6: The rest of the message is read. The operation is posted complete (with or without error) in the ECB. If the reset option was specified and no errors occurred, the reset will be performed.

<u>READ Continue (TT) and READ Continue with Reset (TTR):</u> A READ continue or READ continue with reset macro instruction is issued after a successful read operation to read more data from the same component. The line connection has been established. The channel program for READ TT and READ TTR on the 2740 with dial, transmit control, and checking is:

(1)	Write	Positive response	Note 1
(2)	Read	Data	Note 2
(3)	Write	Ⓓ Ⓒ (TTR only)	
(4)	Disable	(TTR only)	

Note 1: A positive response Ⓨ is sent to indicate that the previous message block was received successfully.

Note 2: The data is received. The data received may be a Ⓒ which terminates the transmission or a message block. The operation is posted complete (with or without error) in the ECB. If the reset option was specified and no errors occurred during transmission, the Ⓓ indicates a good transmission and the Ⓒ turns the terminal motors off.

<u>READ Repeat (TP) and READ Repeat with Reset (TPR):</u> The READ repeat and READ repeat with reset macro instructions are issued, if the previous transmission occurred with error, to reread the message block. The channel program for READ TP or READ TPR is:

(1)	Write	Negative response	Note 1
(2)	Read	Data	
(3)	Write	ⒹⒸ (TPR only)	
(4)	Disable	(TPR only)	

Note 1: A negative response Ⓝ is sent to indicate that the previous message was received in error. The data is then read.

<u>READ Conversational (TV) and READ Conversational with Reset (TVR):</u> The READ TV and READ TVR macro instructions are issued after a write operation. This allows the CPU to read data without re-establishing initial contact with the terminal. The channel program for READ TV and READ TVR is:

(1)	Write	Ⓒ Ⓒ Ⓒ	Note 1
(2)	Write	Selection characters	Note 2
(3)	Read	Response	Note 3
(4)	Read	Data	
(5)	Write	Ⓓ Ⓒ (TVR only)	
(6)	Disable	(TVR only)	

Note 1: Three Ⓒ's are sent.

Note 2: The selection characters (/ and space) are sent so that the terminal can begin to transmit the data.

Note 3: The response is read. If the response is data, then the channel program will continue to read with command 4.

<u>READ Skip (TS):</u> The READ skip macro instruction allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by a user-supplied count. The channel program is a single CCW which is a read command with the skip bit on.

<u>CPU to Terminal</u>

The CPU can either condition the line to receive a call from the CPU or call the terminal. The CPU controls the receiving status of the terminal.

<u>WRITE Initial (TI) and WRITE Initial with Reset (TIR):</u> The WRITE initial and WRITE initial with reset macro instructions are issued to send a message to a terminal. When the CPU calls the terminal, the format of the DFTRMLST is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST, dialcount-integer, dialchar-decchars

The dialcount is the number of digits in the phone number of the terminal and the dialchar is the phone number of the terminal. The channel program when WRITE TI or WRITE TIR points to a call list is:

(1)	Disable		Note 1
(2)	Dial	Dial digits	Note 2
(3)	Write	Pad characters	Note 3
(4)	Write	Data	Note 4
(5)	Read	Response	Note 5
(6)	Write	Ⓒ (TIR only)	Note 6
(7)	Disable	(TIR only)	

Note 1: The line is disabled.

Note 2: The CPU dials the terminal using the number specified in the terminal list.

Note 3: Fifteen pad characters are sent.

Note 4: The response to the LRC comparison, either positive or negative, is read. If the reset option was specified and no errors occurred, the Ⓒ is sent to turn the terminal motors off and the line is disabled.

The terminal may also dial the CPU. The CPU will enable the line so that the call can be received. The format of the DFTRMLST is:

Name	Operation	Operand
symbol	DFTRMLST	DIALST,0

The 0 specifies that this is an answer list. The channel program when WRITE TI or WRITE TIR point to an answer list is:

(1)	Disable		Note 1
(2)	Enable		Note 2
(3)	Write	Pad characters	
(4)	Write	Data	
(5)	Read	Response to VRC/LRC	
(6)	Write	Ⓒ (TIR only)	
(7)	Disable	(TIR only)	

Note 1: The line is disabled.

Note 2: The line is enabled so that the CPU may receive a call from the terminal. When the call is received, this command chains to (3).

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The WRITE continue and WRITE continue with reset macro instructions are issued after read or write operations to send a message back to a terminal. The line connection has previously been established. The channel program for WRITE TT and WRITE TTR is:

(1)	Write	Data
(2)	Read	Response to VRC/LRC
(3)	Write	Ⓒ (TIR only)
(4)	Disable	(TTR only)

WRITE Conversational (TV) and WRITE Conversational with Reset (TVR): The WRITE TV and WRITE TVR macro instructions are issued after a successful read operation to send data. This data could be in response to the data read. The line connection has already been established. The channel program for WRITE TV and WRITE TVR on the 2740 with dial, transmit control, and checking is:

(1)	Write	Ⓓ	Note
(2)	Write	Data	
(3)	Read	Response to VRC/LRC	
(4)	Write	Ⓒ (TVR only)	
(5)	Disable	(TVR only)	

Note: The Ⓓ is a positive response to the previous data read.

WRITE Positive Acknowledge and Disconnect (TA): The WRITE TA macro instruction is issued to send a positive response to the data read and then to disconnect the line. The channel program for WRITE TA is:

(1)	Write	Ⓓ Ⓒ
(2)	Disable	

WRITE Negative Acknowledge and Disconnect (TN): The WRITE TN macro instruction is issued after a read operation when an error occurred during transmission or after a write operation. When issued after a read operation, it is a negative response because no Ⓓ is sent. After a write it is to disconnect the line. The channel program for WRITE TN is:

(1)	Write	Ⓒ
(2)	Disable	

2740 WITH STATION CONTROL

The station control feature of the 2740 provides the terminal with the ability to be addressed or to be polled from the CPU. The CPU does all the necessary polling and addressing. If there is no transmission for a fifteen-second time period, a time-out occurs and the terminal is put into standby status. The terminals are polled or addressed with a one-character identification code.

Terminal to CPU

When the CPU polls the terminals to see if any have data to send, it uses one of the following formats of the DFTRMLST:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST}, {WRAPLST} ({xx-hexchars,}...)
symbol	DTFRMLST	{SSALST }, {SSAWLST} {xxyy-hexchars,}...

Each terminal on the line is polled with a single-character code which identifies the terminal. The xx is the hexadecimal representation of the polling character as it appears in the six-bit BCD transmission code (see Appendix B). The yy is the hexadecimal representation of the space character.

READ Initial (TI) Using OPENLST or WRAPLST: The READ initial macro instruction is issued to begin polling on a line and to read a message. The channel program for READ TI is:

(1)	Write	Ⓒ Ⓒ Ⓒ	Note 1
(2)	Write	Polling character	Note 2
(3)	Write	Space character	Note 3
(4)	Read	Response	Note 4
(5)	Read	Data	Note 5

Note 1: The Ⓒ Ⓒ Ⓒ is sent.

Note 2: The polling character is sent to poll the terminal. The character is the one specified by the DFTRMLST macro.

Note 3: After the polling character, a space character is sent.

Note 4: A two-byte response is read into the input area. If the response is negative (Ⓝ), the channel program is re-initialized at the write polling character command to poll the next terminal. If there is no response within fifteen seconds, a time-out occurs and the channel program is re-initialized to poll the next terminal. If the end of list (OPENLST) is reached before a positive response (Ⓓ followed by text) is received, the operation is posted complete. When a positive response is received, this command chains to (5) to read data.

Note 5: The data is read into the input area.

READ Initial (TI) Using Start-Stop Auto Poll List: The channel program using an open start-stop Auto Poll list (SSALST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	NOP	Note 3
(4)	Read response	Note 4
(5)	Read message	Note 5

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which halts the channel program.

Note 3: This is the last command executed when all negative responses are received.

Note 4: This command is executed on a positive response to bring the index byte (corresponding to the responding terminal) and one text byte into the input area.

Note 5: The messaqe is read into the input area until an EOT is received. The operation is posted as complete in the ECB.

READ Initial (TI) using Start-Stop Wraparound Auto Poll List: The channel program using a start-stop wrap-around Auto Poll list (SSAWLST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	TIC (5)	Note 3
(4)	TIC (7)	Note 4
(5)	Poll first-entry	Note 5
(6)	TIC (5)	Note 6
(7)	Read response	Note 7
(8)	Read text	

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which restarts polling.

Note 3: On a negative response to polling, at the end of the list, this TIC command is executed to start the second Poll command.

Note 4: On a positive response to polling, this TIC command is executed to start the READ response command.

Note 5: If either Poll command terminates with negative response at the end of the list, this command restarts polling at the beginning of the list.

Note 6: On a negative response to polling at the end of the list, this TIC command is executed to restart Poll command (5).

Note 7: Commands (7) and (8) operate the same as commands (4) and (5) of READ initial using SSALST.

<u>READ Skip (TS):</u> The READ skip macro instruction is issued to allow the user to recover from a lost-data error condition. The rest of the data is read from the terminal to clear the line, but it is not received in main storage. The read command reads the amount specified by a user-supplied count. The channel program for READ TS is a single CCW with a read command and the skip bit on.

<u>CPU to Terminal</u>

When the CPU wants to send data to the terminal, it uses the following format of the DFTRMLST:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xx-hexchars,}...)

Each terminal is addressed with a single character code. The xx is the hexadecimal representation of the addressing character as it appears in the six-bit BCD transmission code (see Appendix B).

<u>WRITE Initial (TI) and WRITE Initial with Reset (TIR):</u> The WRITE initial and WRITE initial with reset macro instructions are issued to address a terminal and to send a message. In addition, WRITE TIR resets the line. The channel program for WRITE TI and WRITE TIR is:

(1)	Write	Ⓒ Ⓒ Ⓒ Ⓢ	Note 1
(2)	Write	Addressing character	Note 2
(3)	Write	Space character	Note 3
(4)	Read	Response	Note 4
(5)	Write	Data	Note 5
(6)	Write	Ⓒ Ⓒ Ⓒ (TIR only)	

Note 1: The Ⓒ Ⓒ Ⓒ is sent to set the terminal in standby mode. Ⓢ denotes that addressing is to be done.

Note 2: The addressing character specified in the DFTRMLST is sent to the terminal.

Note 3: The space character is sent after the addressing character.

Note 4: The response to addressing is read into the first byte of the DECB response field. If the response is positive (Ⓨ), the channel program chains to the write command to send the data. If the response is negative (Ⓝ), the operation is posted complete with error in the ECB.

Note 5: The data is sent to the terminal. If the reset option was specified and no errors occurred during transmission, this command chains to (6). Otherwise, the operation is posted complete (with or without error) in the ECB.

<u>Note</u>: When the Autopoll feature is used, the channel program generated for a WRITE Initial for a 2740 does not write the Space character that has to follow the addressing character. This Space character is represented in hexadecimal as yy and must be placed in the xxyy field of the DFTRMLST macro used to define the addressing list. The following format of the DFTRMLST is used when the Autopoll feature is used:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars,} ...)

2740 WITH STATION CONTROL AND CHECKING

The station control feature of the 2740 provides the CPU with the ability to poll or to address a terminal. The terminals are polled and addressed with a single-character identification code. The checking feature consists of the VRC (vertical redundancy check) and the LRC (longitudinal redundancy check). The VRC checks each character as it is sent for correct parity. The LRC checks each message block for correct transmission.

Terminal to CPU

The CPU polls each terminal with a single character that is unique for that terminal on that line. The DFTRMLST used is:

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST}, {WRAPLST} ({xx-hexchars,}...)
symbol	DFTRMLST	{SSALST }, {SSAWLST} {xxyy-hexchars,}...

The xx is the hexadecimal representation of the polling character as it appears in the six-bit BCD transmission code (see Appendix B). The yy is the hexadecimal representation of the space character.

<u>READ Initial (TI) and READ Initial with Reset (TIR) Using OPENLST or WRAPLST:</u> The READ initial and the READ initial with reset macro instructions are issued to start polling or restart polling on a line and to read a message from a remote terminal. In addition, READ TIR sends a positive response Ⓓ and turns the terminal motors off. The channel program for READ TI and READ TIR is:

(1)	Write	Ⓒ Ⓒ Ⓒ	Note 1
(2)	Write	Polling character	Note 2
(3)	Write	Space character	Note 3
(4)	Read	Response	Note 4
(5)	Read	Data	Note 5
(6)	Write	ⒹⒸⒸⒸ (TIR only)	Note 6

Note 1: The Ⓒ Ⓒ Ⓒ is sent to put the terminals in standby mode.

Note 2: The polling character is sent to poll the terminal.

Note 3: A space character is sent.

Note 4: The response to polling is read into the first two bytes of the input area. If the response is negative (Ⓝ) and the end of the list has not been reached, the channel program is re-initialized at command (2) to poll the next entry in the list. If the end of the list is reached before a positive response (Ⓓ and first text character) is received, the operation is posted as complete in the ECB. If a positive response is received, command (6) chains to command (5).

Note 5: The message is read into the input area until an EOB is received. The operation is then posted as complete (with or without error) in the ECB. If the reset option was specified and no errors occurred during transmission, the reset option is performed.

Note 6: The Ⓓ is sent to indicate that the read operation occurred without error. The Ⓒ Ⓒ Ⓒ turns the terminal motors off.

<u>READ Initial (TI) and READ Initial with Reset (TIR) Using Start-Stop Auto Poll List:</u> The channel program using an open start-stop Auto Poll list (SSALST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	NOP	Note 3
(4)	Read response	Note 4
(5)	Read message	Note 5
(6)	Write ⒹⒸⒸⒸ (TIR only)	Note 6

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which halts the channel program.

Note 3: This is the last command executed when all negative responses are received.

Note 4: This command is executed on a positive response to bring the index byte (corresponding to the responding terminal) and one text byte into the input area.

Note 5: The message is read into the input area until an EOB is received.

The operation is posted as complete in the ECB. If READ TIR is specified and no error occurred during transmission, command (5) chains to command (6). Otherwise, the channel program is terminated.

Note 6: On a successful poll, Ⓓ is sent to indicate that the read operation occurred without error. The ⒸⒸⒸ turns the terminal motors off.

READ Initial (TI) and READ Initial with Reset (TIR) using Start-Stop Wraparound Auto Poll List: The channel program using a start-stop wraparound Auto Poll list (SSAWLST) is:

(1)	Write EOT sequence	Note 1
(2)	Poll starting-entry	Note 2
(3)	TIC (5)	Note 3
(4)	TIC (7)	Note 4
(5)	Poll first-entry	Note 5
(6)	TIC (5)	Note 6
(7)	Read response	Note 7
(8)	Read text	
(9)	Write ⒹⒸⒸⒸ (TIR only)	

Note 1: The EOT sequence (ⒸⒸⒸ) is sent to put all terminals on the line in control mode.

Note 2: The starting entry (or next entry) polling characters are sent to poll an entry in the list. On positive response, command (2) chains to command (4). If the end of the list is reached without a positive response, command (2) chains to command (3), which restarts polling.

Note 3: On a negative response to polling, at the end of the list, this TIC command is executed to start the second Poll command.

Note 4: On a positive response to polling, this TIC command is executed to start the Read response command.

Note 5: If either Poll command terminates with negative response at the end of the list, this command restarts polling at the beginning of the list.

Note 6: On a negative response to polling at the end of the list, this TIC command is executed to restart Poll command (5).

Note 7: Commands (7)-(9) operate the same as commands (4)-(6) of READ initial using SSALST.

READ Continue (TT) and READ Continue with Reset (TTR): The READ continue and READ continue with reset macro instructions are issued after a successful READ TI or READ TT to read a subsequent message block from the same component without repolling. The channel program sends a positive response character and chains to a read command. In addition, the reset sends a positive response Ⓓ to the data just read and turns the terminal motors off. To restart polling on a line, another READ TI must be issued. The channel program for READ TT or READ TTR is:

(1)	Write	Ⓨ	
(2)	Read	Data	
(3)	Write	ⒹⒸⒸⒸ	(TTR only)

READ Repeat (TP) and READ Repeat with Reset (TPR): The READ repeat and the READ repeat with reset macro instructions are issued to reread a message block which had completed with error. The channel program writes a negative response (Ⓝ) then chains to a read command.' In addition, the reset sends a positive response (if this read completed without error) and turns the terminal motors off. The channel program for READ TP and READ TPR is:

(1)	Write	Ⓝ	
(2)	Read	Data	
(3)	Write	ⒹⒸⒸⒸ	(TPR only)

READ Skip (TS): The READ skip macro instruction generates a read command with the skip bit on. This macro allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by a user-supplied count. The channel program for READ TS is a single read skip CCW.

CPU to Terminal

When the CPU wants to send data to the terminal, it uses the following format of the DFTRMLST:

Name	Operation	Operand
symbol	DFTRMLST	OPENLST,(xx-hexchars)

Each terminal is addressed with a single-character code. The xx is the hexadecimal representation of the addressing character as it appears in the six-bit BCD transmission code (see Appendix B).

<u>WRITE Initial (TI) and WRITE Initial with Reset (TIR):</u> The WRITE initial and the WRITE initial with reset macro instructions are issued to address a terminal and to send a message in addition to checking to see if transmission was good. WRITE TIR also resets the line if the transmission was good. The channel program for WRITE TI and WRITE TIR is:

(1)	Write	Ⓒ Ⓒ Ⓒ Ⓢ	Note 1
(2)	Write	Addressing character	Note 2
(3)	Write	Space character	Note 3
(4)	Read	Response	Note 4
(5)	Write	Ⓓ	Note 5
(6)	Write	Data	Note 6
(7)	Read	Response to LRC	Note 7
(8)	Write	Ⓒ Ⓒ Ⓒ (TIR only)	Note 8

Note 1: The Ⓒ Ⓒ Ⓒ is sent followed by the Ⓢ which denotes that an addressing function is to be performed.

Note 2: The addressing character is sent.

Note 3: The space character is sent after the addressing character.

Note 4: The response to addressing is read into the first byte of the DECB response field. If the response is positive (Ⓨ), this command chains to a write command (5). If the response is negative (Ⓝ), or if no response is received within fifteen seconds, the operation is posted complete with error in the ECB.

Note 5: A Ⓓ is sent to trigger the checking on the write operation.

Note 6: The message is sent to the terminal.

Note 7: The response to VRC/LRC is read. If the response is positive and the reset option was specified, this reset is performed. Otherwise, the operation is posted complete (with or without error) in the ECB.

Note 8: The Ⓒ Ⓒ Ⓒ is sent to turn the terminal motors off.

<u>Programming Note:</u> Multi-block messages should not be sent to the 2740 M2 with the record checking and buffer receive features. The message transmitted to the 2740 M2 is not printed until a Ⓒ is received, so the WRITE TIR operation should be used with the 2740 M2, unless an EOT is sent to the terminal after a WRITE TI operation.

When a nonproductive operation completion code (X'54') is posted for a WRITE TI or WRITE TIR operation to the 2740 M2, the printer on the terminal is busy, the printer is out of paper, or the terminal is in the local mode. The operation should be retried after waiting a suitable period of time for the printer to complete its operation.

<u>WRITE Continue (TT) and WRITE Continue with Reset (TTR):</u> The WRITE continue and WRITE continue with reset macro instructions are issued after a successful write or read operation to send a message block. In addition, WRITE TTR resets the line if the response from the terminal to VRC/LRC was positive. The channel program for WRITE TT and WRITE TTR is:

(1)	Write	Data
(2)	Read	Response to VRC/LRC
(3)	Write	Ⓒ Ⓒ Ⓒ (TTR only)

<u>WRITE Positive Acknowledge (TA):</u> The WRITE TA macro instruction is issued to send a positive response and to turn off the terminal motors.

(1) Write Ⓓ Ⓒ Ⓒ Ⓒ

<u>WRITE Negative Acknowledge (TN):</u> The WRITE TN macro instruction is issued either after a nonsuccessful read operation or after a write operation. After the read, the terminal interprets this as a negative response, and Ⓒ Ⓒ Ⓒ turns the terminal motors off. After the write operation, this macro instruction resets the line. The channel program for WRITE TN is:

(1) Write Ⓒ Ⓒ Ⓒ

IBM 2260 DISPLAY STATION-IBM 2848 DISPLAY CONTROL (REMOTE)

Terminal to CPU

Message transmission from a 2260 Remote is initiated by the CPU through a READ macro instruction which polls with a two character code. The first character selects the 2848 Display Control; the second character selects the 2260 Display Station.

Name	Operation	Operand
symbol	DFTRMLST	{OPENLST}, {WRAPLST} ({xxyy-hexchars,}...)

xxyy is the hexadecimal representation of the polling characters (see Appendix B). This list may be an open list or a wrap-around list.

The format of a message received through a READ macro instruction is:

STX	device address	[text]	[CAN]	ETX	LRC

STX Start-of-Text character.

device address Character which identifies the sending unit (Display Station or Printer).

text The printer status message has a zero-length text.

CAN A wrong-parity character sent to force an error condition only if the display control detects an internal operation error when transmitting the message.

The user can issue a WRITE erase (TS) macro instruction to clear the error.

ETX End-of-text character.

LRC Longitudinal redundancy check character. The LRC accumulation is started by the STX character and terminated by the ETX character.

READ Initial (TI) - Display or Printer: The problem program issues a READ initial to start or restart polling on a line or to request a printer status. The channel program generated is:

(1)	Write	EOT	Note 1
(2)	Write	Characters	Note 2
(3)	Write	READ MI code	Note 3
(4)	Read	Response	Note 4
(5)	Read	Data	Note 5

Note 1: The EOT character resets the IBM 2848 Display Control address selection to control mode, non-select status, and causes all 2848s on the line to prepare to receive a polling sequence.

Note 2: The polling characters are sent to the terminal, specifying a specific poll of a display station, a general poll of a display control, or a request of a printer status.

Note 3: The command code (READ MI) is sent. This control unit code specifies that the function of the 2848 Display Control is to transmit data from the polled 2260 Display Station. This is one of 5 control unit codes that are unique to the 2848 Remote. These control unit codes specify the function that is to be performed by the 2848 Display Control in connection with the 2260 Remote Display Station.

Note 4: The response to the polling sequence is read. If the response is negative and the end of an open polling list is reached, the operation is posted as complete-with-error in the ECB. If the response is negative and a wraplist is specified, polling is restarted. On a positive response, command (4) chains to command (5).

Note 5: The message is read into main storage. The operation is posted as complete (with or without error) in the ECB.

Specific Poll of Display Station: On positive response (STX), (4) chains to (5) and the message is read. On negative response (EOT) an interruption occurs. BTAM detects the polling restart TP Code, initializes the channel program to poll the next entry within the list, and returns control to the supervisor.

Request of a Printer Status: If the printer is ready and the buffer is empty, a reservation is set on the printer buffer which prevents transmission of messages from the display stations to the printer buffer. If a message is received indicating these conditions, (4) chains to (5). The problem program issues a WRITE continue to print the message. The next EOT resets the reservation condition.

A negative response is either NAK, which indicates the printer is not ready, or EOT, which indicates the printer is ready but the buffer is not empty. If an NAK response is received, the problem program may send a message to an operator. Both negative responses set the printer request condition by which the Display Control, upon receipt of a General Poll, can sense if the printer is ready and the buffer is empty.

General Poll of a Display Control: In this entry, the polling list must specify a general poll; the second byte is a hexadecimal FF. If the printer has a status pending as a result of a previous request (printer status or WRITE initial), this message is transmitted and command (4) chains to command (5).

If the printer is not ready or no request has been made, the display stations are scanned for a message. If a message is pending, it is sent. If there is no message waiting for transmission, a negative response, EOT, is received. The channel program is interrupted; BTAM detects the polling Read response to TP Code, updates the channel program to poll the next entry, and returns control to the supervisor.

READ Continue (TT): The READ continue macro instruction may be issued after a successful READ TI, TT, or TP to read another message block into main storage. This macro should be issued after a general poll. The channel program is:

(1)	Write ACK	Note 1
(2)	Read message	Note 2

Note 1: A positive response is sent to the station acknowledging the message successfully received via the previous READ.

Note 2: The message is read into the user's I/O area.

READ Repeat (TP) Display Only: READ repeat is issued for retransmission of a message. The channel program is:

(1)	Write	NAK	Note 1
(2)	Read	Data	Note 2

Note 1: The write sends a negative response (NAK) to the display station.

Note 2: The message which was received in error is reread.

READ Buffer (TB) - Display Only: The READ buffer is used primarily for diagnostic applications. No polling takes place. The information received is the entire buffer of the specified display station. The channel program is:

(1)	Write	EOT sequence	Note 1
(2)	Write	Polling characters	Note 2
(3)	Write	Code	Note 3
(4)	Read	Data	Note 4

Note 1: The EOT characters reset the 2848s on the line and prepare them to receive a polling sequence.

Note 2: The polling characters are sent to select the 2260 display station whose buffer is to be read. This is a modified form of polling and may be considered a form of selection because the display station is not being polled to see if it has any messages to transmit, but is really being selected so that its buffer in the 2848 can be read.

Note 3: The command code (READ BUFFER) is sent. This control unit code specifies that the function of the 2848 Display Control is to transmit the buffer of the 2260 Display Station indicated.

Note 4: The message is read into main storage. The operation is posted as complete (with or without error) in the ECB.

The user should issue a WRITE erase after READ buffer.

READ Skip (TS): The READ skip macro instruction generates a read command with the skip bit on. This macro allows the user to recover from a lost-data error condition. The rest of the message is read from the terminal to clear the line, but it is not received in main storage. The command reads the amount specified by a user-supplied count. The channel program for READ TS is a single read skip CCW.

CPU to Terminal

Transmission from the CPU to the 2260 Remote is initiated by the CPU through a WRITE macro instruction. Two characters are used to address.

Name	Operation	Operands
symbol	DFTRMLST	OPENLST, ({xxyy-hexchars,}...)

xxyy is the hexadecimal representation of the addressing characters (see Appendix B). xx represents the 2848 address character and yy the device address character.

The format of a message sent through a WRITE macro instruction is:

STX	text	ETX	LRC

WRITE Initial (TI) - Display or Printer: The problem program issues a WRITE initial macro instruction to send the first block of a message. The channel program is:

(1)	Write	EOT sequence	Note 1
(2)	Write	Characters	Note 2
(3)	Write	WRITE code	Note 3
(4)	Read	Response	
(5)	Write	Message block	
(6)	Read	Response	

Note 1: The EOT character resets the IBM 2848 Display Control address selection to control mode, non-selected status, and causes all 2848s on the line to prepare to receive an addressing sequence.

Note 2: The characters are addressing characters which indicate a a 2848 and a display station or printer.

Note 3: The command code (WRITE) is sent. This control unit code specifies that the function of the 2848 Display Control is to receive data.

If a printer is addressed, command (4) reads the addressing sequence response. A negative response (EOT or NAK) interrupts the channel program. The NAK, which indicates that the printer is not ready, or EOT, which indicates the printer is ready but the buffer is not empty, sets a printer request.

If the response is positive (ACK), indicating that the printer is ready and available, command (4) chains to command (5) and the message is sent. If a transmission error occurs, the operation is stopped, and the printer buffer is cleared. The user may retry transmission by issuing a WRITE continue.

If a display station is addressed, (4) reads the addressing sequence response which is normally positive (ACK), and command (4) chains to command (5). If a transmission error occurs, the user may retry by issuing a WRITE continue but the erroneous message will not be cleared. The user may issue a WRITE erase to resend the same message, or a READ buffer and WRITE erase if several messages were displayed.

WRITE at Line Address (TL) - Display only: The channel program issues a WRITE at line address macro instruction to position the cursor to start on a specified line. Characters are displayed starting from that point. The channel program is:

(1)	Write	EOT
(2)	Write	Addressing characters
(3)	Write	WRITE LINE code
(4) - (6)	same as WRITE initial	

If a transmission error occurs, the user may try sending the same message again with a WRITE continue. Data will be displayed starting at the same line.

Note: The following values are the possible line address values:

2848 /2260 Character	Hexadecimal Code Representation	Display Line
0	X'50'	1
1	X'51'	2
2	X'52'	3
3	X'53'	4
4	X'54'	5
5	X'55'	6
6	X'56'	7
7	X'57'	8
8	X'58'	9
9	X'59'	10
:	X'5A'	11
;	X'5B'	12

WRITE Erase (TS) - Display only: The problem program issues a WRITE erase macro instruction to erase the screen and to display a message starting at the upper left hand corner. The channel program is:

(1)	Write	EOT
(2)	Write	Addressing characters
(3)	Write	ERASE code
(4) - (6)	same as WRITE initial	

If a transmission error occurs, the user may issue another WRITE erase to send the same message.

WRITE Continue (TT) - Display or Printer: The problem program may issue a WRITE continue macro instruction after any READ/WRITE operation where the option reset is not specified. The channel program is:

(1)	Write	Data
(2)	Read	Response

If the previous operation was a WRITE initial addressing a printer, the WRITE continue macro instruction retries transmission of the message. If the previous operation was a WRITE at line address macro instruction or a WRITE at line address macro followed by one or several WRITE continues, the cursor is positioned at the beginning of the line specified in the message. If a transmission error occurs, the user may retry by issuing WRITE continue. In all other cases, the characters are displayed on the station starting at the cursor position. When text has been received, an answer may be provided by a WRITE continue without re-selecting the display.

WRITE with Reset (TIR, TLR, TSR, TTR): All of the previously described WRITE macro instructions can be issued with reset. The additional command sends an EOT to set the line in control mode, non-select status. If the data is received in error, the reset function is not performed and the line is not reset.

- WRITE initial with reset (TIR) establishes line connection, sends the first message block, and resets the terminal.
- WRITE at line address with reset (TLR) positions the cursor, displays a message, and resets the terminal.
- WRITE erase with reset (TSR) erases the CRT, displays the message, and resets the terminal.
- WRITE continue with reset (TTR) operates the same as WRITE continue and resets the terminal.

WRITE Positive Acknowledge (TA) - Display or Printer: When a message is received, the problem program may issue a WRITE positive acknowledge macro instruction to indicate to the sending station that the message was received without error and to end the operation. The channel program for WRITE TA is:

```
(1)  Write    STX
(2)  Write    EOT
```

WRITE Negative Acknowledge (TN) - Display or Printer: When a message in error is received, the problem program may issue a WRITE negative acknowledge macro instruction. This indicates to the sending station that the message is not correct and ends the operation. The channel program for WRITE TN is a single CCW which writes an EOT.

IBM 2260 DISPLAY STATION-IBM 2848 DISPLAY CONTROL (LOCAL)

General Information

The 2260 Local differs in operation from other BTAM-supported devices because it is locally attached. The 2260 Local is not polled on read operations. Instead, the 2260 operator keys in the START symbol and text and then depresses the ENTER key in order for the message to be read. Depressing the ENTER key results in a CPU I/O interrupt with the Attention bit ON in the status. This I/O interrupt is referred to as an Attention interrupt in the following discussion.

When an Attention interrupt occurs at the CPU, a CCB (a part of the LCB in BTAM) for that line address must be in the channel scheduler queue in order for the Attention status to be posted (see Communication Between the Problem Program, BTAM, and the Supervisor). If a CCB is not in the queue, the Attention interrupt is ignored by the DOS supervisor.

BTAM queues each display station LCB (including the required CCB) associated with the DTFBT when the first READ Initial (or READ Short) is issued in the problem program. After queuing each LCB, BTAM returns control to the problem program. Subsequently, when an Attention interrupt occurs, the BTAM Interruption Handler requests the supervisor to start a Read for that line address. When the message is received from the 2260, BTAM posts the READ macro complete. The LCB for the 2260 just read is dequeued while the other LCB's in the DTFBT remain queued. However, no further Attention interrupts will be serviced until the next READ macro is issued. Only one message is read per READ macro. Attention interrupts occurring between READ's are posted in the appropriate LCB's for service on subsequent READ's.

The rln operand has a special use in READ initial (or READ short). This operand specifies which LCB (i.e., 2260) in the DTFBT is to be checked first for an Attention interrupt. BTAM scans all LCB's starting with the one specified in the rln operand. During the scan, BTAM checks for previously posted (but not yet serviced) Attention interrupts. Servicing of an Attention interrupt consists of reading from the 2260 that caused it. Any LCB found to be dequeued during the scan is again queued. The scan is terminated when BTAM services an LCB having an Attention posted or when all LCB's have been scanned at least once. Once a READ initial (or READ short) is issued, the problem program is prevented from issuing another READ or WRITE for that DTFBT until the first READ completes or until a RESETPL is issued to terminate the first READ.

The use of the CHGNTRY macro with the SKIP option allows the user to cause BTAM to ignore a particular 2260. An LCB for a "skipped" 2260 will not be queued during a scan. If it is already queued when CHGNTRY is issued with SKIP specified, a subsequent Attention interrupt caused by the associated 2260 will not be honored. CHGNTRY with ACTIVATE specified will reactivate an LCB. That is, if the LCB is currently dequeued, it will be queued during the scan for a subsequent READ initial (or READ short). If it is currently queued, an Attention interrupt caused by the 2260 will be serviced during the current or subsequent READ.

The other 2260 Local READ/WRITE options (READ buffer, WRITE initial, WRITE at line address, and WRITE erase) do not involve the scanning of the LCB's. For these options the requested operation is performed on the specific display station (or the printer) specified via the <u>rln</u> operand. Upon completion of these macros, the specified LCB is dequeued. CHGNTRY has no effect on the execution of these macros.

When issuing one of these latter macros following a RESETPL that terminated an incomplete READ initial (or READ short), the user should be aware of the remote possibility of losing data. RESETPL neither halts currently executing commands nor dequeues LCB's. It is used to prevent servicing of Attention interrupts occurring subsequent to its execution. Hence, the issuance of a WRITE initial, for example, may require that BTAM first dequeue a queued LCB and stop current I/O activity by a supervisor-issued Halt I/O instruction. Only then could the Write CCW be issued for that 2260 (or 1053). If a Read CCW started prior to RESETPL had not yet terminated at the time WRITE initial was issued, the read operation on the channel would be halted. The 2848, however, would proceed to a normal ending of the operation at the 2260. The clearing of the START symbol (for READ initial) or the advancing of the cursor (for READ short) on the 2260 screen would occur as on a successful read operation.

<u>Terminal to CPU</u>

<u>Read Initial (TI)</u>: The READ initial macro instruction is issued to read a message from anyone (but only one) of the non-skipped display stations defined in the DTFBT. A READ TI causes the same channel program (a single Read CCW) to be moved into each non-skipped LCB in the DTFBT.

When the operator of the 2260 depresses the ENTER key and the attention interrupt is recognized by the CPU, all characters displayed between the START symbol and the cursor, except the characters to the right of any NL symbols, are read into main storage.

- The START symbol indicates the beginning of the message. After the message is read, the START symbol is cleared from the screen automatically. If no START symbol has been keyed, no message is read.
- NL stands for new line. The characters to the right of an NL symbol are not read.
- The cursor indicates the position on the screen following the last character.

A READ initial macro instruction allows the CPU to read any one of the 2260's defined in the DTFBT when the operator presses the ENTER key. When the READ initial is completed, the relative position in the DTFBT of the 2260 that sent the message is stored in the low order byte of the DECB polling pointer field.

<u>Note</u>: If the area operand is coded 'S', the I/O address is that of the buffer (only one buffer is received). The length of the message to be read must be less than buffer length-12. The entry operand is not used for the 2260 Local.

<u>READ Short (TO)</u>: The READ short macro instruction is similar to READ initial, except the START symbol is not cleared and the cursor is advanced one position after the Read.

<u>READ Buffer (TB)</u>: The READ buffer macro instruction causes the entire buffer of the specified display station to be read and the cursor to be placed in the top left corner. If the nondestructive cursor feature is not installed, the user should next issue a WRITE erase (TS) to clear the buffer.

The relative line number operand for the READ buffer macro instruction specifies the relative position in the DTFBT of the LCB corresponding to the 2260 whose buffer is to be read.

CPU to Terminal

The WRITE operations for the 2260 Local initiate CPU-to-terminal transmission.

WRITE Initial (TI): The WRITE initial macro instruction is issued to transmit data to a specific display station starting at the position on the screen indicated by the cursor. The channel program for WRITE TI is a single CCW that writes the message. If the relative line number operand refers to a 2260, the message is displayed on the screen. If the operand refers to a 1053 printer, the message is typed.

WRITE at Line Address (TL): The WRITE at line address macro instruction is issued to transmit data to a specific display station. The cursor is positioned at the first character position of a line on the screen as determined by the first character transmitted from the user-specified output area. Characters are displayed from that point.

WRITE Erase (TS): The WRITE erase macro instruction is issued to erase the screen of a specific display station and position the cursor in the upper left corner of the screen.

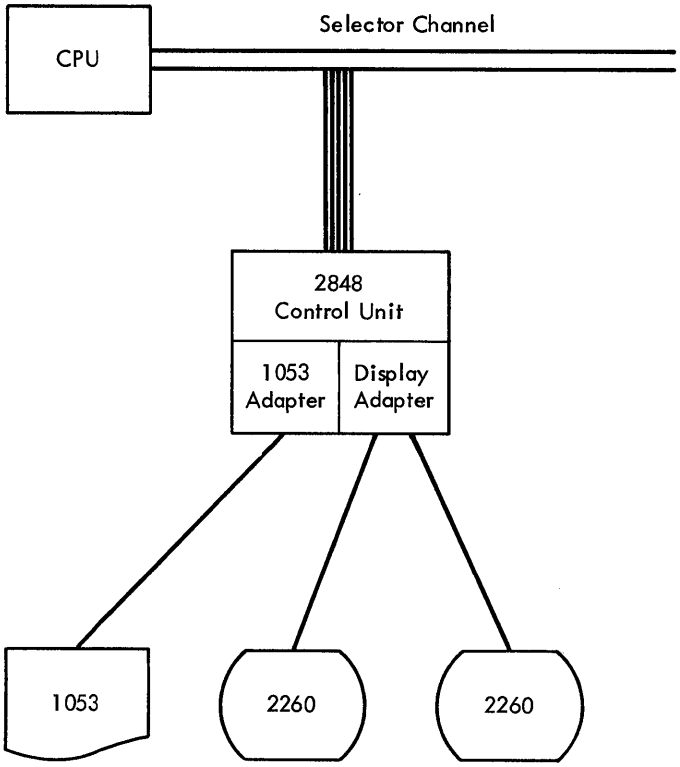
Figure 25 gives the 2260 local configuration.

Programming Note: If a READ initial or READ short is outstanding for the DTFBT, if no terminal has responded, and if the user wishes to initiate a different operation, he must first issue a RESETPL macro specifying the same DECB specified in the incomplete READ macro. The RESETPL should be followed by a WAIT macro.

Lock Option

BTAM provides the 2260 Local user with a Lock option for each operation type. The user may request this option by adding the suffix "L" to the normal operation type codes, yielding the additional macro options:

READ TIL
READ TOL
READ TBL
WRITE TIL
WRITE TLL
WRITE TSL



Note: The number of Display Adapters which may be included in a single 2848 Display Control varies.

Figure 25. 2260 Local Configuration

Each standard operation (without the Lock option) restores (unlocks) the manual entry keyboard when the operation is complete. Each operation with the Lock option functions like the corresponding standard operation, except that at the completion of the READ or WRITE operation, the manual entry keyboard retains the locked or unlocked status that existed at the beginning of the execution of the operation. If the manual entry keyboard is unlocked prior to the execution of a READ TBL, WRITE TIL, WRITE TLL, or WRITE TSL operation, the keyboard will be unlocked at the completion of the operation. If the manual entry keyboard is locked prior to the execution of a READ TBL, WRITE TIL, WRITE TLL, or WRITE TSL operation, the keyboard will remain locked at the completion of the operation; this prevents further operator entry until the ERASE key or the RESTORE key is depressed or until a BTAM macro instruction without the Lock option is issued. The READ TIL and READ TOL operations always leave the manual entry keyboard locked. WRITE commands with the Lock option should be used in conjunction with READ commands with the Lock option. If an error condition is detected as a result of a READ command with the Lock option, a WRITE command with the Lock option should follow the READ command. The locked keyboard gets the operator's attention and insures that the operator reads the message displayed, before further keying on the keyboard.

CHANNEL PROGRAM FOR IBM 7770/7772 AUDIO RESPONSE UNITS

GENERAL INFORMATION

INPUT MESSAGES: Input messages received by the IBM Audio Response Unit are transferred to main storage as an 8-bit representation of the telephone company A B C code (except those messages dialed from IBM 3944 terminals which are transferred to the processor directly in EBCDIC representation). Input messages are interpreted by the user's program to format an appropriate response.

OUTPUT MESSAGES: Output messages from the System/360 are converted to audio response by the 7770 or the 7772, and routed to the telephone terminals that originated the inquiries.

The 7770 accepts an output message in the form of drum addresses that are used to locate specific words stored on the 7770 audio drum.

On the 7772, the output message contains digitally coded voice words, each word consisting of variable number of bytes. The bytes representing a voice word are usually stored in a direct access storage device and the user must read it into main storage before sending the message.

The user is advised against adding pause words at the end of an audio message that is to be followed by a Read (as in WRITE Invitational, or WRITE Conversational) so that the caller would not be liable to enter the inquiry while still in the Write operation. This would cause an overrun and loss of data. On the other hand, he should add one or more pause words at the end of an audio message to be followed by a disconnect (reset option or CONTROL Disable) in order to avoid the click of the disconnect being heard right after the last word.

VOCABULARY: The 7770 provides a basic vocabulary of 32 words. With an expansion of words in groups of 16, the 7770 can offer a maximum vocabulary of up to 128 words per unit. The 7772 vocabulary is limited only by the capacity of the direct access storage allocated to it in the system.

LINES: The 7770 lines are offered in groups of 4; up to a maximum of 48 lines can be attached to each 7770.

The 7772 lines are offered in groups of 2; up to a maximum of 8 lines can be attached to each 7772.

However, the 7772 offers one output channel (voice code translator) per pair of lines. A VCT is not assigned to any specific pair of lines. When an output message is sent to a line, the 7772 hardware automatically switches this line to the first available VCT. If none are available, the Write command is stacked in the 7772 control unit until a VCT is freed.

CONFIGURATION: The IBM 7770 and IBM 7772 Audio Response Units are attached to a multiplexor channel. They present to the channel a unique interface regardless of the data sets and the terminals to which they are connected. One exception is the IBM 7772 (for those pairs of lines with the dial terminal feature that are connected to the IBM 3944 World Trade Terminal) that sends the input bytes in EBCDIC code instead of the audio response unit special code. Since code conversion is the user's responsibility, the handling of these two control units is independent of the terminals used. Therefore, any reference to the 7770 and 7772 also implies the terminals to which they are connected.

PROGRAMMING INFORMATION

Polling and Addressing Lists

Since the IBM 7770/7772 terminals cannot be polled or addressed, the following macro instructions must not be used with audio response units:

DFTRMLST
RESETPL
CHGNTRY

The ENTRY operand in the WRITE macro instruction will not point to an entry in an addressing list. Instead it will be used to specify the address where the input message should be read when the conversational or invitational options are used. The maximum length of the input message is specified in the MSGL operand of the DTFBT macro instruction.

Buffer Management

Buffer management is not permitted because it does not serve any useful purpose with audio response units. Because these units have real time requirements, the user cannot afford to wait for a buffer to be released from the pool on a read operation. On a READ initial, the enabling of the line would be delayed, thus leaving it inoperative. On a write operation when the buffer pool is empty, the audio answer would be delayed. This is undesirable because the inquirer is waiting for the answer. For this reason an Audio application program should set aside all the I/O area it needs rather than use a buffer pool.

FEATURE

Only the FEATURE=(PCI) option of the DTFBT macro instruction operand is allowed for the IBM 7772. No FEATURE option is supported for the IBM 7770. As previously explained, the IBM 7772 has one output channel (VCT) for each two input lines. The user may use as many output areas as he has input lines. In this case he does not need the PCI feature. On the other hand, never more than half the number of output areas are involved at one time in an output operation. The user may decide to use only as many output areas as he has output channels. No problem arises if the conversation optype is not used. If it is used, the need arises for a method of using the output area during the time used by an input operation. This is performed by the PCI feature. When the P-bit is posted in the DECB after a WRITE conversational macro instruction, it signals that the Read operation has started and, consequently, that the output buffer used by the previous Write operation is no longer used and can be filled with an audio message for another line. Thus, main storage space is saved while retaining an efficient usage of the VCTs.

Programming Note: A PCI option is provided only for the 7772. When the operand FEATURE=(PCI) is specified in the DTFBT macro instruction, a PCI bit is set in the read CCW. If the PCI occurs alone, no completion code is set, byte 0 of the ECB is 0, and bit 0 of the DECB flag byte is set to 1.

The BTAM multiple wait routine considers a PCI occurring alone as one event. When a PCI satisfies a wait, the user must set bit 0 of the flag byte to 0 before re-issuing the wait.

READ Macro Instructions

READ Initial (TI): READ initial enables the line to receive a call. Once the call has been received, READ initial will read the inquiry message. The channel program is:

(1)	Enable		Note 1
(2)	Read	Data	Note 2

Note 1: The line is enabled to receive a call.

Note 2: The inquiry message is read.

WRITE Macro Instructions

WRITE Initial (TI) and WRITE Initial with Reset (TIR): A WRITE initial macro instruction enables the line to receive a call. Once the call has been received, a specified audio message can be sent to the caller. The channel program for WRITE TI is:

(1)	Enable		Note 1
(2)	Write	Message	Note 2
(3)	Disable (TIR only)		Note 3
(4)	TIC to (1) (TIR only)		Note 4

Note 1: The line is enabled to receive a call. When the call is received, command (1) chains to command (2).

Note 2: A specific audio message is sent to the caller. When reset is specified and no error occurs, the channel program is re-initialized automatically and no posting is made. This loop is stopped when the line group is closed. The same procedure is followed in case the user hangs up. If an error occurs, the line will be disconnected and the error will be posted. No re-initialization takes place after an error.

Note 3: The line is disabled.

Note 4: The TIC points to the enable command thus allowing the line to be enabled.

The purpose of the WRITE initial macro instruction is to provide the 7772 user with a means to send the first segment of a multisegment audio message when the complete data for an audio message cannot be loaded into main storage. A READ macro instruction must not follow the WRITE TI, since the audio response units require that a Read CCW be command-chained to the previous CCW to avoid possible loss of input characters. If the user needs a WRITE initial followed by a read sequence, he may use the WRITE invitational macro instruction.

When specified with reset, the WRITE initial macro instruction implements the information mode. When a user dials the Audio Response Unit, no inquiry is expected; rather a fixed format audio message (for example, a weather forecast) is expected. Then the line is disconnected and made ready for another call. To change the mode in which the line is used (for instance, to return to the inquiry mode or to change or update the contents of the audio message), the line group must be closed, the change made, and the line opened again, unless no further activity is required for the line group.

WRITE Invitational (TC): A WRITE invitational macro instruction enables the line to receive a call. When the call has been received, WRITE TC sends the specified audio message and reads the inquiry. The channel program for a WRITE TC macro instruction is:

(1)	Enable		Note 1
(2)	Write	Message	Note 2
(3)	Read	Inquiry	Note 3

Note 1: The line is enabled to receive a call. When the call is received, command (1) chains to command (2).

Note 2: The audio message is sent to the caller.

Note 3: The input message is read into main storage.

WRITE Continue (TT) and WRITE Continue with Reset (TTR): The WRITE continue allows the user to send an audio message in multiple segments. The last segment can be sent with a WRITE continue with reset to disconnect the line at the end of the transaction. The channel program for WRITE TT is a single Write command. The channel program for WRITE TTR is a Write command followed by a Disable command.

WRITE Conversational (TV): The WRITE conversational macro instruction allows the user to send an audio message and to read the next inquiry. The channel program is the same as steps 2 and 3 of WRITE TC. If FEATURE=(PCI) is coded for the 7772, the PCI flag is ON in the Read CCW.

CONTROL Macro Instructions

CONTROL Initial (TI): A CONTROL initial macro instruction allows the user to enable the line to receive a call. When a user calls the CPU, the connection is established and the operation is posted as complete. The channel program for CONTROL TI is a single enable command.

The facility allows the 7772 user to prepare the audio message only after the connection has been established.

CONTROL Disable (TD): The CONTROL disable macro instruction allows the user to disconnect the line when the transaction has been completed. The channel program for CONTROL TD is a single disable which disconnects the line.

The CONTROL disable macro instruction must be used whenever the reset option cannot be used.

CPU-TO-CPU CONTENTION

General Information

Point-to-point Binary Synchronous Communication is supported by BTAM over a nonswitched line between the CPU and a remote CPU that is also a System/360 using BTAM support.

The CPUs operate in contention. On rare occasions each CPU may simultaneously bid for control of the line by executing a WRITE initial type macro instruction at the same time. Using the facilities of BTAM, the user creates a programmed system to break contention. The problem programs in both CPUs must be coordinated so that one is programmed to act as the master and the other as the slave in the contention situation only. This is accomplished via the FEATURE keyword operand in their respective DTFBT macro instructions. One must have coded MAS in the FEATURE sublist and the other must have coded SLV. (See DTFBT Macro Instruction.)

Transmission is initiated when one of the CPUs successfully sends the Enquiry signal (ENQ character) to the other CPU. Because polling and addressing are not used in this application, the entry operand in the READ/WRITE macro instructions is not used except for READ TTL, READ TPL, WRITE TQ, and the Conversational-type WRITEs.

Note: The user must provide all text framing characters (SOH, STX, US, ETB, ETX, DLE STX). BTAM provides DLE ETB, DLE ETX, and all other control characters.

READ Macro Instructions

There are eight options available for the READ macro instruction. The optypes used are:

TI - READ Initial

TIQ - READ Initial Inquiry

TT - READ Continue

TTL - READ Continue with Leading Graphics

TP - READ Repeat

TPL - READ Repeat with Leading Graphics

TQ - READ Inquiry

TTS - READ Stop

Only normal completions are described for the channel programs discussed in the following sections. The section Completion Codes discusses posting of irregular completions.

Read Initial (TI): This macro instruction is issued to monitor the line for the Enquiry signal (ENQ) from the remote CPU. When this signal is received and acknowledged, the first message from the remote CPU is read into the specified input area. The channel program generated and executed for READ initial is:

(1)	Prepare to read	Note 1
(2)	Read response	Note 2
(3)	Write response	Note 3
(4)	Read message block	Note 4

Note 1: Initialize the adapter to listen for the ENQ character from the remote CPU.

Note 2: Read character from remote CPU into the first byte of the response field in the DECB. If this character is the ENQ character, restart the channel program to write the positive response and to read a message.

Note 3: Write positive acknowledgment (ACK-0).

Note 4: Read message block into the I/O area and post the operation complete in the ECB.

READ Initial Inquiry (TIQ): This macro instruction permits the problem program (at initial contact time) to monitor the line for a request to transmit from the remote CPU. When the ENQ character with which the remote CPU is bidding for the line is detected, the operation is postèd complete. If the user does not wish to read data, he may write the Wait-before-transmitting (WABT) characters (WRITE TW), or if he wishes to receive the data he may issue a READ TT macro instruction. The channel program is:

(1)	Prepare to read	Note 1
(2)	Read ENQ character	Note 2

Note 1: This command initializes the adapter to listen for a transmission from the remote CPU.

Note 2: This command reads the ENQ character from the remote CPU into the first byte of the response field in the DECB.

The READ TIQ macro (like READ TI) reinitializes the alternating acknowledgment information used by BTAM to check that the appropriate alternating acknowledgments are sent and received.

READ Inquiry (TQ): This macro instruction may be used when the CPU, as the receiving station, times out on a text-Read command in a READ macro channel program. As the receiving station, the CPU does not initiate recovery from such a time out but generally issues the READ inquiry macro awaiting an ENQ from the remote CPU (transmitting station) requesting the CPU to transmit its last response. For this reason READ inquiry differs from READ initial inquiry (TIQ) in that READ inquiry does not cause BTAM to reinitialize the information on alternating acknowledgments. When ENQ is received, the appropriate macro can then be executed to retransmit the last response of the CPU.

The channel program generated and executed is a single command:

(1) Read the ENQ into DECB Response Field.

Note: The user should issue this macro only if he desires more retries than are provided automatically by BTAM ERP.

READ Continue (TT): This macro instruction is issued to read in messages from the remote CPU after initial contact has been successfully established. The channel program is:

(1)	Write positive acknowledge	Note 1
(2)	Read message block	Note 2

Note 1: Write the appropriate alternating acknowledgment.

Note 2: Read a message block. The data received will be either the next block of data or an EOT character.

READ Continue with Leading Graphics (TTL): This macro instruction is used to perform the same basic function as READ continue, i. e., acknowledge the last message received and read the next message into main storage. In addition, however, this macro permits the sending of a maximum of 7 graphic characters (i.e., non-control characters) preceding the BTAM-supplied positive acknowledgment sequence (either ACK-0 or ACK-1). The leading graphics must be supplied by the user in an output area of the following format:

1. The first byte (count byte) of the output area must contain a binary count of the graphics to be sent. This count must not be zero. A count of zero results in a channel program check due to an invalid count.

2. The bytes following the count byte must contain the graphics in transmission code.

The user must point to the count byte via the entry operand in the macro instruction. While the user may place the count byte and graphics in a buffer, the main storage address specified by the entry operand must be that of the count byte itself.

Note: Leading graphics can serve any function required by the application program. The only restriction is that the bit configuration of the graphics not coincide with the bit configuration of defined data link control characters (e.g., ENQ, STX, etc).

Read Repeat (TP): A READ TP is issued following an unsuccessful READ TI or READ TT to request retransmission of the last block of data from the remote CPU. The channel program is:

(1)	Write negative acknowledgment	Note 1
(2)	Read message block	Note 2

Note 1: Write an NAK to the remote CPU indicating the last message block was received in error.

Note 2: Read a message block.

Note: This macro should be issued only if the user desires more retries than BTAM ERP has already attempted automatically.

READ Repeat with Leading Graphics (TPL): This macro instruction is used to perform the same function as READ repeat but in addition allows the user to send leading graphics with the BTAM-supplied negative acknowledgment (NAK). As explained for READ continue with leading graphics, the user specifies the address of a count byte via the entry operand. The graphics follow the count byte in the output area. The maximum number of graphics that may be sent is seven.

Note: For READ TTL and READ TPL, BTAM ERP does not automatically request retransmission when data check and overrun occur on received messages in intermediate block check mode. (See CONTROL macro.)

READ Stop (TTS): This macro instruction is issued to allow the receiving CPU to request the transmitting remote CPU not to send any additional message blocks. The SAK character sequence is transmitted to the remote CPU, indicating that the CPU has received the previous message block but does not wish to receive additional message blocks. The transmitting CPU should unconditionally respond by transmitting an EOT (or optionally, DLE EOT on a dial line) to the CPU to yield control of the line. The channel program is:

(1) Write SAK character sequence Note 1
(2) Read response Note 2

Note 1: Write Stop Acknowledgment character sequence (SAK).

Note 2: Read response from the CPU into the Response field in the DECB (bytes 26 and 27).

WRITE Macro Instructions

There are 13 options available for the WRITE macro instruction. The optypes used are:

TI - WRITE Initial

TIE - WRITE Initial Transparent Block

TIX - WRITE Initial Transparent Text

TIXV - WRITE Initial Transparent Conversational

TIV - WRITE Initial Conversational

TT - WRITE Continue

TE - WRITE Transparent Block

TX - WRITE Transparent Text

TXV - WRITE Transparent Conversational

TV - WRITE Conversational

TR - WRITE End-of-Transmission

TW - WRITE Wait-before-Transmitting (WABT)

TQ - WRITE Inquiry

The five WRITE initial macro instructions are capable of breaking contention, that is, they establish the direction of transmission between the two CPUs on a point-to-point uncontrolled line. To establish the direction of transmission, the CPU wishing to transmit bids for the line by transmitting an ENQ character, and then reading the response. If the proper alternating acknowledgment (ACK-0) is received, the CPU wishing to transmit has successfully gained the line and may transmit data.

A true contention situation arises when both CPUs issue a WRITE initial at the same time. In this case, one CPU must act as the master and the other, as the slave. The master or slave status of a CPU is specified by the user by coding MAS or SLV, respectively, in the sublist of the FEATURE keyword operand of the DTFBT macró instruction.

When contention does occur, the Feature field in the DTFBT is examined by BTAM to determine the recovery procedure to be used. BTAM immediately ends the WRITE initial for the slave CPU with completion code 50. The slave CPU problem program should issue the READ initial or READ Initial Inquiry macro. Meanwhile, BTAM retries the bidding sequence automatically for the master CPU. Should the retries be unsuccessful, BTAM posts completion code 44 and ends the WRITE initial. Figure 26 lists the various completion codes posted when various error conditions occur on WRITE initial. Retry options are also specified. For a description of irregular completions of channel programs, see the section, Completion Codes.

CPU	Completion Code	Meaning	Retry Options
Master	44	ENQ, NAK, or I/O Error received in response to the ENQ sent to the remote CPU.	1
	41 with Data Check	(TIV, TIXV only) Message received from remote CPU was in error.	2
	60	Response received to message sent to remote CPU was invalid acknowledgment.	2
	41	NAK received in response to text (Bit 7 of byte 29 in the DECB is set if n retries fail).	4
Slave	50	Bid for line unsuccessful. Master CPU is also bidding for the line.	5
	44	NAK or invalid response received in reply to ENQ.	1
	41 with Data Check	Same as for Master CPU.	2
	60	Same as for Master CPU.	3
	41	Same as for Master CPU.	4

Retry Options

1. Retry the Write Initial-type macro.
2. Initiate a READ Repeat macro.
3. Initiate a WRITE Inquiry macro.
4. Initiate a WRITE Continue-type macro.
5. Issue a READ Initial or a READ Initial Inquiry.

Figure 26. Retry Options for Completion Codes on WRITE Initial in CPU-to-CPU Contention

WRITE Initial (TI): This macro instruction is used to bid for the line and to establish the direction of transmission. The channel program generated and executed is:

(1) Write the ENQ character Note 1
(2) Read the response characters Note 2
(3) Write the message Note 3
(4) Read the response to the message Note 4

Note 1: Send ENQ character to bid for the line.

Note 2: Read response characters into the two byte response field in the DECB. If the response is ACK-0, the channel program is restarted to execute the balance of the channel program.

Note 3: Write a message block.

Note 4: Read the response to checking.

WRITE Initial Transparent Block (TIE) and WRITE Initial Transparent Text (TIX): These two macro instructions are used when the user desires to gain control of the line and send data over the line in transparent mode. BTAM provides either the 2-character End-of-transparent-block sequence (DLE,ETB) for the TIE optype or the End-of-transparent-text sequence (DLE,ETX) for the TIX optype. This means the user is responsible only for the 2-character Start-of-transparent-text (DLE STX) sequence appearing in his message. The length operand specifies only the number of bytes in the user's output area and does not include the end characters provided by BTAM.

The channel program generated and executed for these two macros is:

(1) Write ENQ character Note 1
(2) Read response characters Note 1
(3) Write message block Note 2
(4) Write termination characters Note 3
(5) Read response to checking Note 1

Note 1: The functions of these commands are identical to those listed for the corresponding commands in WRITE TI.

Note 2: Send the message block. Data transmission is terminated by count or buffer length, not by an end character.

Note 3: Send the end characters. For WRITE TIE the end characters are DLE, ETB; for WRITE TIX the end characters are DLE ETX.

<u>WRITE Initial Conversational (TIV):</u> This macro instruction is used when the user desires to bid for the line, send a message to the remote CPU, and receive a message in response. Since the CPU controls the direction of transmission by successfully bidding for the line, line discipline permits the remote CPU to send a message only if the CPU is prepared to receive that message. The CPU controlling the line can only receive the first message from the remote CPU by means of one of the Conversation-type WRITE's. WRITE initial conversational provides this facility at initial contact time.

It is the user's responsibility to coordinate the mutual exchange of messages (conversation) between the CPUs within the problem programs for the two devices. This is properly a function of message processing and is not performed by BTAM.

In coding any one of the Conversational-type WRITE's, it is required that the user specify his output message via the area and length operands, and define the equivalent parameters for the expected input message by specifying the address of a parameter list via the entry operand. The parameter list contains 2 fullwords with the format shown in Figure 27.

Word 1		Address	of Input	Area	(right adjusted)
Word 2		Length	of Input	Area	(right adjusted)

Figure 27. Parameter List for Conversational WRITE's

If the address specified in Word 1 is zero, it is assumed that the user is requesting BTAM to read the input message into buffers. Word 2, in this case, must specify the maximum length of the expected message. BTAM performs the buffering in the same manner as discussed for the READ macro instruction when <u>area</u> is coded 'S' in that macro. The only exception is that the address of the first buffer (into which the received message is read) is placed in Word 1 of the parameter list rather than into the <u>area</u> field of the DECB.

The channel program generated and executed for this macro is:

(1) Write ENQ character Note 1
(2) Read response characters Note 1
(3) Write message block Note 1
(4) Read response Note 2

Note 1: The functions of these commands are identical to those of the corresponding commands in WRITE TI.

Note 2: Response characters are read into an input area pointed to via the entry operand in the macro. The response may be either the appropriate alternating acknowledgment or a message.

There are certain restrictions on the use of this and the other Conversational-type WRITE's.

- The first Conversational-type WRITE issued by the CPU controlling the line permits the other CPU to send its first message. Whether this is the TIXV, TIV, or TV type is the user's option. Coordination of the conversation between the CPUs is a function of message processing and not of BTAM.
- The user is cautioned that a Conversational-type WRITE should be followed by a READ continue in normal operation (no errors) and never by another WRITE if a message (rather than ACK-0 or ACK-1) has been received on the Conversational-type WRITE. This restriction must be followed in order to safeguard against losing a message without being able to recover.
- The message sent, via WRITE conversational, must end with ETX and never ETB, since ETB implies more blocks are to be sent. This is a rule of the line control procedures.
- An additional use of the Conversational-type WRITE's is to receive a positive or negative acknowledgment with leading graphics in response to the message transmitted. Such a response can only be received into a user-specified input area since the normal response field (in the DECB) is only 2 bytes. If leading graphics with NAK are received, ERP will not automatically retransmit the previously sent message. It is not the function

of BTAM to analyze leading graphics. Instead this function is part of the user's application. (See the CONTROL macro for a discussion of this point.)

WRITE Initial Transparent Conversational (TIXV): This macro combines the functions of types TIX and TIV, in that transparent text may be sent to the remote CPU and a message response received. The channel program is:

(1)	Write ENQ character	Note 1
(2)	Read response	Note 1
(3)	Write message	Note 2
(4)	Write ending characters	Note 3
(5)	Read response	Note 4

Note 1: The function of these commands is identical to that of the corresponding commands in WRITE TI.

Note 2: Write the message of transparent text. The Write command is stopped by count or buffer length, rather than end characters.

Note 3: Write the ending characters, DLE ETX.

Note 4: Read the response. It may be either an appropriate positive acknowledgment or a message. (See WRITE Initial Conversational for the format of the parameter list pointed to via the entry operand.)

WRITE Continue (TT): The channel program executed for WRITE TT is:

(1)	Write message block	
(2)	Read response	Note

Note: The alternating positive acknowledgment is read into the two byte response field in the DECB.

WRITE Transparent Block (TE) and WRITE Transparent Text (TX): The channel program executed for these WRITE types is:

(1)	Write message block	Note 1
(2)	Write ending characters	Note 2
(3)	Read response characters	Note 3

Note 1: Write a block of transparent text terminated by count.

Note 2: Write ending characters:
DLE ETB for WRITE TE
DLE ETX for WRITE TX

Note 3: Read alternating positive acknowledgment into DECB response field.

WRITE Conversational (TV): The WRITE conversational is issued after initial contact has been established. It enables the CPU to receive a message in response to a message sent.

The channel program generated and executed is:

(1)	Write message block
(2)	Read response into user's input area

Note: The response may be an appropriate alternating acknowledgment or a message block. The acknowledgment will be in the input area. If an acknowledgment is received instead of a message, it is checked by BTAM for correctness and completion code 7F posted. (See WRITE Initial Conversational for the format of the parameter list pointed to via the entry operand.)

WRITE Transparent Conversational (TXV): The channel program generated and executed is:

(1)	Write message block	Note 1
(2)	Write ending characters	Note 2
(3)	Read response	Note 3

Note 1: Write a message block terminated by count.

Note 2: The ending characters will be DLE ETX.

Note 3: The response is read into the input area; it may be an appropriate alternating acknowledgment. (See WRITE Initial Conversational for the format of the parameter list pointed to via the entry operand.)

WRITE Inquiry (TQ): This macro instruction is issued when no response or an invalid response is received from the remote CPU in reply to a message.

In either case, the user must specify the address of an input area into which the acknowledgment or response message is to be read. The user must specify both the input area and length in a parameter list as described for WRITE Initial conversational (TIV). The entry operand must specify the address of the parameter list. A completion code of 7F indicates normal completion (recovery). The acknowledgment, if received, is checked by BTAM for validity.

The channel program generated and executed is:

(1) Write ENQ Note 1
(2) Read response Note 2

Note 1: The ENQ character is written from the table of special characters.

Note 2: The response characters are read into the user-specified input area.

Note: This macro should be issued only when the user desires more retries than BTAM ERP automatically provides (the number of retries being equal to that specified in the RETRY operand in DTFBT).

WRITE Wait-Before-Transmitting (TW): This macro instruction is issued by the problem program to inform the remote CPU that a temporary condition exists, requiring a delay before the remote CPU continues transmission. This may result, for example, from a temporary lack of buffers in the CPU for receiving messages from the transmitting CPU.

This macro instruction may be issued following completion of a READ initial inquiry (TIQ) which had been issued by the slave CPU in recovering from the contention situation (See Figure 26).

This macro instruction may also be issued, after a message has been read, to defer sending a negative or positive response. Again, this could be for the reason that some temporary condition exists, making immediate reading of a message undesirable.

Because the WRITE WABT is used in place of a READ continue, READ repeat, or (if the remote CPU had sent its last message by a WRITE Conversational-type macro) a WRITE by the problem program, the remote CPU generally responds with an ENQ requesting that the CPU (which sent the WABT) transmit the deferred response. For this reason, the channel program for WRITE WABT includes the Prepare and Read commands chained to the Write command sending the WABT.

The channel program is:

(1) Write WABT Note 1
(2) Prepare the line to receive ENQ Note 2
(3) Read ENQ Note 3

Note 1: Write the 2-character sequence for WABT.

Note 2: Prepare the line to monitor the line.

Note 3: The Prepare chains to the Read when the remote CPU sends the ENQ. The ENQ character is read into the response field of the DECB.

When this macro is posted complete, the normal completion code is 7F because the ENQ is expected and is not sent in lieu of text. Text messages are illegal responses to a WABT. Within the discipline of line control procedure, the only response expected, other than possibly an end-of-transmission signal, is the ENQ. After completion, the user may re-issue the WRITE WABT macro, if still not ready, or he may issue the appropriate macro to send his deferred response or message.

WRITE End-of-Transmission (TR): This macro instruction is issued to inform the remote CPU that transmission is ended on the line. Generally, this macro is issued by the problem program which controls the line discipline. It may, however, be used by the problem program to end transmission when irrecoverable errors exist.

The channel program generates and executes a Write command which sends the End-of-Transmission character (EOT). This places the line in control mode. Further transmission on the line requires that the READ initial/WRITE initial be used to re-establish contact between the CPUs.

Use of the CONTROL Macro Instruction

For CPU-to-CPU contention, only the CONTROL Mode (TM) is available for use. Enabling and disabling of the line are performed by the OPEN and CLOSE macro instructions for nonswitched lines. The CONTROL macro is not available for these functions.

CPU-TO-CPU DIAL

General Information

Binary Synchronous Communication is supported by BTAM over a switched network between the CPU* and a remote CPU* that is also a System/360 using BTAM support.

Transmission is initiated when either the CPU or the remote CPU performs a calling operation. For this reason, both

----

*Throughout the discussion on CPU-to-CPU Dial, the convention followed is that "CPU" means the System/360 in which the user's problem program resides while "remote CPU" is used to refer to the remote System/360 with which the problem program is communicating.

calling and answering functions are provided for the CPU by BTAM. Optional Identification (ID) Verification procedures are provided by BTAM in lieu of polling/addressing procedures, which are not used for CPU-to-CPU (Dial) communication.

When transmission is initiated by either the CPU or the remote CPU, there is a standard basic line control procedure which is always employed, regardless of the ID Verification options also used:

1. The calling CPU dials and sends the Enquiry character, ENQ.

2. The called CPU answers, reads the ENQ character, and responds with a positive acknowledgment (ACK-0) if the line connection is valid, or with no response if the line connection is not valid. No response is sent by the called CPU, for example, when the ENQ character is received with I/O error. The calling CPU would then retry sending the ENQ character.

3. On a positive response to the ENQ, the calling CPU transmits the first message.

ID Verification Procedures

There are three optional BTAM-supported ID Verification procedures, which provide the user with greater circuit assurance and security protection.

I Am/Who Are You? (IAM/WRU): Using this procedure, both the calling CPU and the called CPU exchange ID graphic characters when line connection is being established:

1. The calling CPU dials and sends the ENQ preceded by a maximum of 15 graphic characters to the called CPU, as proof of identity.

2. The called CPU answers, reads the ID-ENQ sequence, verifies it, and if the ID is:

   a. valid, sends the ACK-0 characters, preceded by a maximum of 15 ID graphic characters of its own, to the calling CPU.

   b. invalid, does not reply. Instead ERP restarts the channel program at the Read CCW, awaiting the retry of the ID,ENQ by the calling CPU.

   c. invalid, after n retries (n = number specified by RETRY in DTFBT), may issue a WRITE Disconnect or a CONTROL Disable to break the line connection.

3. The calling CPU then:

   a. verifies the ID-ACK-0 response sequence and if valid, transmits the message. If the ID sequence is invalid, the calling CPU responds by retrying its ID-ENQ sequence again or by sending the disconnect signal, depending on the user's option.

   b. if the response is NAK, may either retry the ID-ENQ sequence or send the disconnect signal.

   c. if the response is the disconnect signal (DLE, EOT), disables the line by issuing the CONTROL Disable macro.

I Am (IAM): This procedure requires that the calling CPU send the ID graphic characters preceding the ENQ and that the called CPU verify and send only a positive acknowledgment (ACK-0) in response to a valid ID sequence. Invalid ID's are handled in the same manner as discussed for IAM/WRU.

Who Are You (WRU): This procedure requires that the calling CPU send the ENQ character, only, and that the called CPU respond with an ID sequence preceding ACK-0. Recovery is similar to that discussed for IAM/WRU.

Prerequisites: The user chooses the ID Verification procedures desired by coding the appropriate codes in the sublist of the FEATURE keyword operand in the DTFBT macro instruction as shown in Figure 28.

Procedure	Code for Calling Operation	Code for Answering Operation
IAM	SIX	RIX
WRU	SXW	RXW
IAM/WRU	SIW	RIW

Figure 28. ID Verification Codes

The code is dependent only upon what procedure is to be employed by the calling CPU. The code for the calling operation at the calling CPU must correspond to the code for the answering operation at the called CPU. For example, if the CPU is calling the remote CPU and employing the IAM procedure, the problem program must code

SIX for the calling operation in the FEATURE sublist. Similarly, it is necessary that the remote problem program, requesting the answering operation in the remote CPU, have coded RIX in its FEATURE sublist. Since the CPU can both call and answer calls from the remote CPU, it is possible that in calling the procedure would be IAM and in answering, WRU. The user would code:

FEATURE=(BSC,SIX,RXW)

The user program at the remote CPU would correspondingly code:

FEATURE=(BSC,RIX,SXW)

If one of the calling/answering ID codes is omitted in the sublist, ID verification is not performed for the relevant operation (calling or answering). If the calling/answering codes are omitted completely, this means the user has chosen not to use ID Verification procedures for either calling or answering operations on that line.

Note: It is the user's responsibility to know which procedure, if any, is expected by the remote CPU.

READ Macro Instruction

There are six types of READ macro instructions available. The "optypes" used are:

TI - READ Initial

TT - READ Continue

TTL - READ Continue with Leading Graphics

TP - READ Repeat

TPL - READ Repeat with Leading Graphics

TQ - READ Inquiry

For the channel programs described in the following sections, only normal completions are fully described. The section Completion Codes describes posting of irregular completions.

READ Initial (TI): This macro instruction may be used to execute either the answering function (the remote CPU is calling the CPU) or the calling function (the CPU is calling the remote CPU). In either case, however, the problem program issues the READ initial for the purpose of receiving a message from the remote CPU. Thus, the execution of the calling function via READ initial is different from the execution of the calling function via WRITE initial. The latter macro is issued by the problem program for the express purpose of transmitting a message to the remote CPU (see the section on WRITE initial).

When performing the answering function, READ initial answers an incoming call, verifies the received ID (IAM,IAM/WRU), sends an ID sequence (WRU,IAM/WRU) preceding ACK-0, and reads in the message block received from the remote CPU.

If no ID Verification is used (that is, none of the codes, RIX, RXW, or RIW, is coded in the FEATURE sublist of the DTFBT for the line group), the entry operand in the READ initial must specify the address of a terminal list constructed by the following macro.

Name	Operation	Operands
symbol	DFTRMLST	DIALST,0

The 0 specifies an answer list.

When ID Verification procedures are used, it is necessary that the user specify the address, via the entry operand, of an IDLST type terminal list defined by a DFTRMLST macro instruction in the problem program.

Name	Operation	Operands
symbol	DFTRMLST	IDLST,0,numrec-integer, [idrec-hexchars,] numsent-integer, [idsent-hexchars]

IDLST and 0 must be coded as shown. The 0 indicates an answer list. The numrec operand specifies the number of ID characters expected from the calling terminal. This would be zero for the WRU procedure. For the IAM and IAM/WRU procedures this number would be equal to the number of ID characters plus one (for ENQ character). The idrec operand is omitted for WRU. For IAM and IAM/WRU it specifies the hexadecimal representation (the X' ' framing characters are not coded) of the transmission code for the ID characters plus the ENQ character. Thus, if four ID characters and an ENQ are expected, numrec would specify 5 and idrec would be specified by 10 hexadecimal digits.

The numsent operand is 0 for the IAM procedure. For WRU and IAM/WRU it specifies the number of ID characters to be sent to the remote (calling) CPU. The number specified does not include the ACK-0 characters, since they are supplied by BTAM. For IAM the idsent operand is omitted. For WRU and IAM/WRU the idsent operand specifies the hexadecimal representation (without X' ') of the transmission code for the ID characters being sent. If four ID characters are to be sent, numsent would specify 4 and idsent would be coded as the 8 hexadecimal digits representing the ID characters.

If no ID Verification is used, the channel program generated and executed is:

(1) Enable the line to receive a call — Note 1
(2) Read the ENQ — Note 2
(3) Write the ACK-0 response — Note 3
(4) Read the message — Note 4

Note 1: The line is enabled to receive a call.

Note 2: When a call is received, the ENQ character is read into the Response field of the DECB.

Note 3: After the ENQ is read, the appropriate positive acknowledgment is sent.

Note 4: Read the message into the user's input area.

Note: No Disable command is issued at the start, since the user should have issued a CONTROL Disable to perform this function if the line is not currently disabled.

If the WRU procedure is used, the channel program generated and executed is:

(1) Enable the line — See Note 1 (above)
(2) Read the ENQ — See Note 2 (above)
(3) Write ID sequence — Note
(4) Write ACK-0 response — See Note 3 (above)
(5) Read the message — See Note 4 (above)

Note: The characters defined by the idsent operand are sent to the remote (calling) CPU.

If the IAM procedure is used, the channel program generated and executed is:

(1) Enable the line
(2) Read the ID,ENQ into list — Note
(3) Write ACK-0 in response
(4) Read the message

Note: The ID, ENQ sequence from the remote CPU is read into the terminal list and verified. If it is valid, command (3) is executed.

If the IAM/WRU procedure is used, the channel program generated and executed is:

(1) Enable the line
(2) Read the ID,ENQ into list
(3) Write ID sequence
(4) Write ACK-0
(5) Read the message

When performing the calling function, READ initial dials the telephone number of the remote CPU, establishes valid line connection, indicates to the remote CPU that the call was made for the purpose of receiving messages, and then receives the first message block (if any) that the remote CPU transmits.

If the user does not use ID Verification (that is, none of the codes, SIX, SXW, or SIW, is coded in the FEATURE sublist of the DTFBT macro for the line group), the entry operand specifies the address of a terminal list defined via DFTRMLST as follows.

Name	Operation	Operands
symbol	DFTRMLST	DIALST, dialcount-integer, dialchars-decchars

The dialcount specifies the number of dialing characters to be used (example: 7) and dialchars specifies the actual dialing characters (example: 1234567).

The channel program generated and executed for READ initial without ID Verification to perform the calling function is:

(1) Dial the user-specified number — Note 1
(2) Write the ENQ character — Note 2
(3) Read the (ACK-0) response — Note 3
(4) Write the EOT character — Note 4
(5) Read the response character(s) — Note 5
(6) Write ACK-0 — Note 6
(7) Read message — Note 7

Note 1: The user-specified telephone number is dialed.

Note 2: The ENQ character is transmitted to the remote CPU. The ENQ is provided by BTAM.

Note 3: The response to the ENQ is read into the DECB.

Note 4: If the appropriate alternating acknowledgment (ACK-0) is received, the EOT character is transmitted. This indicates to the remote CPU that the problem program is yielding the right-to-transmit to the remote CPU. The EOT character is provided by BTAM.

Note 5: The response from the remote CPU is read into the DECB. Depending on the status of the remote CPU one of the following responses may be received:

a. ENQ - indicates the remote CPU is bidding for the line in order to transmit.

b. EOT - indicates the remote CPU has nothing to send and is relinquishing its right-to-transmit. (The problem program may either bid for the line or send the disconnect signal upon receiving EOT).

c. DLE,EOT - indicates the remote CPU has no message to transmit and is disconnecting. (The problem program should issue a CONTROL macro to disable the line upon receiving DLE,EOT).

Note 6: If ENQ is received, BTAM sends the ACK-0 response to the remote CPU.

Note 7: The first message block from the remote CPU is read into the user-specified area.

If ID Verification is used in the READ initial (one of the codes, SIX,SXW, or SIW, is coded in the FEATURE operand sublist of the DTFBT for the line group), the user must provide, via the entry operand, the address of an IDLST constructed by a DFTRMLST macro.

Name	Operation	Operands
symbol	DFTRMLST	IDLST,dialcount-integer, dialchars-decchars, numrec-integer, [idrec-hexchars,] numsent-integer [,idsent-hexchars]

The nonzero dialcount indicates a calling list. This operand also specifies the number of dialing characters to be used. The dialchars operand specifies the dialing characters to be used. The numrec operand specifies the number of ID characters to be received from the remote CPU. The idrec operand specifies the actual characters expected. For example, if A,B,C,ACK-0 is expected, numrec would be coded as 5 and idrec (EBCDIC transmission code) would be coded C1C2C31070. The numsent operand specifies the number of ID characters to be transmitted to the remote CPU. The idsent operand specifies the actual ID characters to be transmitted. Neither numsent nor idsent includes provision for ENQ, which is supplied by BTAM. For example, if A,B,C,ENQ is to be transmitted, numsent specifies 3, while idsent is coded C1C2C3. Depending upon which procedure (IAM,WRU, or IAM/WRU) is employed, idrec or idsent is optionally coded.

The channel programs generated and executed for READ initial with a calling list and ID Verification are similar to that shown for no ID Verficiation, except that IDs are exchanged.

Figure 29 lists the various completion codes which may be returned for possible errors in READ initial. The listed retry options may be employed when the user desires further attempts to recover without having to disconnect and re-enable the line. However, the user is reminded that BTAM automatically retries all of the listed errors the number of times specified in the retry operand of the DTFBT macro issued for the line group. The completion codes listed are posted by BTAM only if this number of retries fails to recover from the indicated errors.

Macro Used	Completion Code	Meaning	Procedure Used	Retry Options
Read Initial (with answering List)	44	ENQ not received properly	No ID Verif. WRU	1 1
		Received ID, ENQ did not compare to expected ID, ENQ in IDLST	IAM IAM/WRU	1 1
	62	ENQ or ID, ENQ was received instead of the text message, meaning the remote (calling) CPU either did not receive or rejected the positive acknowledgment sequence (ACK-0 or ID, ACK-0) sent.	No ID Verif. WRU IAM IAM/WRU	2 or 5 2S or 5 2 or 5 2S or 5
	41 with lost data, data check, or overrun	Text message received with I/O error.	No ID Verif. WRU IAM IAM/WRU	3 or 5 3 or 5 3 or 5 3 or 5
	41 with time out	No text message received.	No ID Verif. WRU IAM IAM/WRU	1, 4, or 5 1, 4, or 5 1, 4, or 5 1, 4, or 5

Retry Options: The S following the number in the table means the user must code 'S' for the *entry* operand in the retry macro. (BTAM performs the necessary ID Verfication again.)

1. Issue a CONTROL Disable (TD) to disconnect the line.

2. Issue a READ (TT) (2S: Code 'S' for *entry* operand).

3. Issue a READ Repeat (TP).

4. Issue a READ Inquiry (TQ).

5. Issue a WRITE Disconnect (TD) to break the connection.

Figure 29: Error Completion Codes and Retry Options for READ Initial (answering) (Part 1 of 3)

Macro Used	Completion Code	Meaning	Procedure Used	Retry Options
READ Initial (With Calling list)	44	NAK or invalid ID received in response to sent ENQ.	No ID Verif. WRU	1 or 5 1S or 5
	44	NAK or invalid ID received in response to sent ID, ENQ.	IAM IAM/WRU	1S or 5 1S or 5
	60	Wrong acknowledgment received (ACK-1 instead of ACK-0).	No ID Verif. WRU IAM IAM/WRU	1 or 5 1S or 5 1S or 5 1S or 5
	41 with time out	No response received to ENQ or ID, ENQ (the sent ID may be invalid). The Read CCW has a TP code of hex 05 (for No ID or IAM) or hex 07 (for WRU or IAM/WRU).	No ID Verif. WRU IAM IAM/WRU	1 or 5 1S or 5 1S or 5 1S or 5
	41 with time out	No text message received. The Read CCW has a TP Code of hex 11 (no buffers) or hex 12 (buffers).	No ID Verif. WRU IAM IAM/WRU	2, 3, or 5 2, 3, or 5 2, 3, or 5 2, 3, or 5
	41 with lost data, data check, or overrun	Text message received with I/O error. The READ CCW has a TP code of hex 11 (no buffers) or hex 12 (buffers).	No ID Verif. WRU IAM IAM/WRU	4 or 5 4 or 5 4 or 5 4 or 5

Retry Options: The S following the number in the table means the user must code 'S' for the *entry* operand in the retry macro. (BTAM performs the necessary ID Verfication again.)

1. Issue a WRITE Inquiry (TQ) (1S: Code 'S' for *entry* operand).

2. Issue a CONTROL Disable (TD) to disconnect the line.

3. Issue a READ Inquiry (TQ) to monitor the line for an ENQ from the remote station.

4. Issue a READ Repeat (TP).

5. Issue a WRITE Disconnect (TD) to break the connection.

Figure 29: Error Completion Codes and Retry Options for READ Initial (calling) (Part 2 of 3)

Macro Used	Completion Code	Meaning	Procedure Used	Retry Options
Initial-type WRITE'S	44	NAK or invalid ID received in response to sent ENQ.	No ID Verif. WRU	1 or 4 1S or 4
		NAK or invalid ID received in response to sent ID, ENQ.	IAM IAM/WRU	1S or 4 1S or 4
	60	Wrong acknowledgment received in response to ENQ or ID, ENQ. Read CCW has TP code of hex 05 (no ID or IAM) or hex 07 (WRU or IAM/WRU)	No ID Verif. WRU IAM IAM/WRU	1 or 4 1S or 4 1S or 4 1S or 4
	60	Wrong acknowledgment received in response to the transmitted message. Read CCW has a TP code of hex 20	No ID Verif. WRU IAM IAM/WRU	1 or 4 1 or 4 1 or 4 1 or 4
	41	NAK received in response to text message sent. Bit 7 of DECB byte 29 is set.	No ID Verif. WRU IAM IAM/WRU	2 or 4 2 or 4 2 or 4 2 or 4
	41 with lost data, data check or overrun	Text message received with I/O error (TIV,TIXV only).	No ID Verif. WRU IAM IAM/WRU	3 or 4 3 or 4 3 or 4 3 or 4
	41 with time out	No response received to the sent message	No ID Verif. WRU IAM IAM/WRU	1 or 4 1 or 4 1 or 4 1 or 4

Retry Options: The S following the number in the table means the user must code 'S' for the entry operand in the retry macro. (BTAM performs the necessary ID Verification again.)

1. Issue a WRITE Inquiry (TQ) (1S: Code 'S').

2. Issue a WRITE Continue-type macro to retransmit the message.

3. Issue a READ Repeat (TP).

4. Issue a WRITE Disconnect (TD) to break the connection.

Figure 29: Error Completion Codes and Retry Options for Initial-type WRITE'S (Part 3 of 3)

READ Continue (TT): This macro instruction is the same as the READ continue described in the section, CPU-to-CPU Contention except when entry is coded 'S', an ID sequence is retried (see Figure 29).

READ Continue with Leading Graphics (TTL): This macro is also described in the section CPU-to-CPU Contention.

READ Repeat (TP): This macro instruction is used to request retransmission of a message received in error. The channel program is a Write NAK command, chained to a Read text-message.

READ Repeat with Leading Graphics (TPL): See the description of this macro in the section CPU-to-CPU Contention.

READ Inquiry (TQ): This macro instruction is used for the CPU, when it is the receiving station, to recover from a text-Read time out occurring in a previously issued READ macro instruction. The initiative to attempt recovery belongs to the remote CPU (the transmitting station), which sends the ENQ character when its own Read response times-out. When this ENQ is received, the READ inquiry is completed and the user may issue the appropriate macro instruction to allow data exchange to continue.

The channel program generated and executed is a single command:

(1) Read the ENQ into DECB Response Field

Note: The user should issue this macro only if he desires more retries of a text-Read time-out than are provided automatically by BTAM ERP.

WRITE Macro Instructions

There are 14 options available for the WRITE macro instruction. The optypes used are:

TI - WRITE Initial

TIE - WRITE Initial Transparent Block

TIX - WRITE Initial Transparent Text

TIXV - WRITE Initial Transparent Conversational

TIV - WRITE Initial Conversational

TT - WRITE Continue

TE - WRITE Transparent Block

TX - WRITE Transparent Text

TXV - WRITE Transparent Conversational

TV - WRITE Conversational

TR - WRITE End-of-Transmission

TW - WRITE Wait-Before-Transmitting (WABT)

TQ - WRITE Inquiry

TD - WRITE Disconnect

For irregular completions of the channel programs described in the following sections, see the section Completion Codes.

WRITE Initial (TI): This macro instruction is used to call the remote CPU, establish valid line connection, and send a message to the remote CPU. If no ID verification is employed, the user must specify, via the entry operand, the address of a DIALST type terminal list constructed by a DFTRMLST macro instruction in the problem program. If the number to be dialed is 12345678, for example, the user would code:

```
DLIST    DFTRMLST      DIALST,8,12345678
```

The inlist positional operand is used only for WRITE initial conversational (TIV) and WRITE initial transparent conversational (TIXV) and is explained in a later section.

Name	Operation	Operands
symbol	DFTRMLST	DIALST, dialcount-integer, dialchars-decchars [,inlist-relexp]

The channel program generated and executed is:

(1) Dial the digits specified Note 1
(2) Write the ENQ character Note 2
(3) Read the response (ACK-0) Note 3
(4) Write the message from output area Note 4
(5) Read the response to message (ACK-1) Note 5

Note: The channel program does not start out by disabling the line because it is assumed that the user disabled the line previously through a RESETPL, a WRITE Disconnect (in a previous call), or a CONTROL Disable. Execution of a Disable command takes 3 seconds even if the line is already disabled, therefore the channel program does not perform this function.

The distinction between RESETPL and CONTROL Disable is that RESETPL is used to Halt I/O on an Enable which did not complete because no call was received. CONTROL Disable disables a line on which connection was established.

Note 1: The specified phone number is dialed.

Note 2: The ENQ is sent to the remote CPU.

Note 3: The response to the ENQ is read.

Note 4: If the response is correct (ACK-0), the message is sent.

Note 5: The response to the message is read.

If ID Verification is used in the WRITE initial, the user must provide, via the entry operand, the address of an IDLST constructed by a DFTRMLST macro.

Name	Operation	Operands
symbol	DFTRMLST	IDLST,dialcount-integer, dialchars-decchars, numrec-integer, [idrec-hexchars,] numsent-integer [,idsent-hexchars] [,inlist-relexp]

For the IAM/WRU procedure, all operands are coded, except inlist, which is used only for WRITE initial conversational (TIV) and WRITE initial transparent conversational (TIXV).

The numrec and idrec operands refer, respectively, to the number of ID characters expected from the remote (called) CPU and the hexadecimal representation for them. The user is required to include the ACK-0 characters in these operands. This means, if 4 ID characters plus ACK-0 are expected, numrec specifies 6 and idrec is coded as 12 hexadecimal digits, the last 4 of which represent ACK-0. The numsent and idsent operands refer, in analogous manner, to the ID characters to be sent to the remote CPU. They do not include the ENQ character, since this is provided by BTAM. So if 4 ID characters are to be sent preceding the ENQ, numsent specifies 4 and idsent is coded as 8 hexadecimal digits.

The IAM and WRU procedures require similar coding except that for:

- IAM, numrec is coded 0 and the idrec operand is omitted.

- WRU, numsent is coded 0 and the idsent operand is omitted.

The channel program generated and executed for IAM/WRU,IAM, and WRU are similar to that shown for No ID Verification, except ID's are exchanged.

Figure 29 lists the error completion codes and possible retry macros to use to recover without disabling and redialing.

WRITE Initial Conversational (TIV) and WRITE Initial Transparent Conversational (TIXV): The basic function of these macros is the same as for WRITE initial, in that line connection is established by dialing the remote CPU. Aside from ID Verification, which is unique to a switched connection, these macros are similar to the WRITE initial conversational and WRITE initial transparent conversational for CPU-to-CPU Contention (see that section).

However, in the DFTRMLST macro instruction, defining the DIALST (no ID Verification) or the IDLST (IAM,WRU, and IAM/WRU), the user must also code the inlist positional operand. This operand specifies the symbolic address of a 2 word parameter list:

- Word 1 contains the address of the input area provided for the message response from the remote CPU. If Word 1 contains all zeros, BTAM provides buffers for the expected message.

- Word 2 contains the length of the input area.

The other operands are coded in similar manner to that described for the ID Verification procedures in the preceding section on WRITE initial.

WRITE Initial Transparent Block (TIE) and WRITE Initial Transparent Text (TIX): These macros handle line connection as described for WRITE initial and are similar to the equivalent macros discussed for CPU-to-CPU Contention in providing the appropriate end characters for the message being sent to the remote CPU.

Other Macro Instructions: The macro instructions listed are identical to the equivalent macros discussed in the section CPU-to-CPU Contention.

TT - WRITE Continue

TE - WRITE Transparent Block

TX - WRITE Transparent Text

TXV - WRITE Transparent Conversational

TV - WRITE Conversational

WRITE End-of-Transmission (TR): This macro instruction is used to relinquish the right of the CPU to transmit on a switched line. A logical use of this macro instruction is in place of the Conversational-type WRITE's. For example, when both the CPU and the remote CPU have messages to exchange it might be that the CPU would first transmit all of its messages before receiving any from the remote CPU. In this case after sending its last message (ending in ETX) and receiving the appropriate alternating acknowledgment in reply, the CPU could then execute the WRITE End-of-Transmission macro to notify the remote CPU that there are no more messages to be sent. The remote CPU would then be allowed to send ENQ and become the transmitting station.

The channel program generated and executed is:

(1) Write EOT Note 1
(2) Read Response into DECB Response field Note 2

Note 1: The EOT character is sent to relinquish the right to transmit over the line.

Note 2: This command reads the response into the Response Field of the DECB. On a time-out of this command (no response is received), it is retried until a response is received or "n" retries have occurred following the initial attempt. Since the Read command has a 3-second time-out, a total of 3n seconds of line time may be expended in waiting for a response from the remote CPU. Note: "n" is equal to the number specified via RETRY in the DTFBT.

The normal responses from the remote CPU are:

- ENQ indicating the remote CPU wishes to become the transmitting station.
- EOT indicating the remote CPU also has nothing to transmit and relinquishes its right to transmit.
- DLE, EOT indicating the disconnect-signal.

This macro instruction causes BTAM to reinitialize the BTAM-maintained information on alternating acknowledgments. This information is used by BTAM to check that the appropriate alternating acknowledgments are sent and received.

WRITE Wait-Before-Transmitting (WABT) (TW): This macro is the same as explained in CPU-to-CPU Contention. If it is not posted complete (the remote CPU does not respond to the WABT), the user must issue a RESETPL to cancel the incomplete Prepare command. The user could then issue a CONTROL Disable to disconnect the line.

WRITE Inquiry (TQ): This macro may be issued for the following reasons:

1. To send the ENQ character when no response, an invalid response, or a WABT sequence has been received from the remote CPU in reply to a message. This is the same reason it is issued for CPU-to-CPU Contention and means that the user desires more retries than BTAM ERP has already attempted.

2. To send the ENQ character as a bid for the line. It is used in this manner, for example, after EOT has been received from the remote CPU indicating the remote CPU has yielded its right to transmit. (Note that in CPU-to-CPU Contention this function is provided by WRITE Initial.)

3. To recover from certain error conditions which can occur on READ Initial (Calling list) or on Initial-type WRITE'S (see Figure 29). As discussed in (1), issuance of WRITE Inquiry for this purpose indicates the user desires more retries (without re-dialing) than BTAM ERP has already attempted.

In the case of (1) or (2) the user must specify, via the entry operand, the address of a parameter list (see Figure 27) containing the input area address and length. If the correct acknowledgment is received, BTAM posts normal completion. Invalid responses are retried the number of times specified by the user (see RETRY in DTFBT) and appropriately posted if irrecoverable. A NAK response is posted immediately, indicating for case (1) that the previously sent message should be retransmitted or for cases (2) and (3) that the remote CPU is not ready. If a message is received, normal completion is posted.

In the case of (3) the user may code entry as 'S' (see Figure 29). In this case BTAM restarts the previously generated channel program (for the READ or WRITE Initial) at the command following the Dial command. When entry is not coded as 'S' the channel program generated and executed is identical to that discussed for CPU-to-CPU Contention.

WRITE Disconnect (TD): This macro instruction is used on a switched-line data link to inform the remote CPU that disconnection is occurring.

The channel program is:

(1) Write DLE,EOT Note 1
(2) Disable the line Note 2

Note 1: The disconnect signal (DLE,EOT) is sent to the remote CPU.

Note 2: The line connection is broken.

This macro may be issued at normal end of communication between the CPUs (generally by the calling CPU) or when irrecoverable errors occur on the line.

Use of the CONTROL Macro Instruction

Both CONTROL Disable (TD) and CONTROL Mode (TM) may be issued by the problem program for the CPU-to-CPU Dial application. The functions of each type are:

- CONTROL Disable (TD): Used to disable the switched connection when the disconnect signal (DLE,EOT) is received from the remote CPU. The user is alerted to this disconnect signal being received when bit 2 of the Flag Byte in the DECB is on and a completion code of 7F to a READ or 41 to a WRITE (rarely) is set. Generally, the calling CPU sends the disconnect signal since it initiated the call by dialing. The disconnect signal is sent via the WRITE Disconnect (TD) macro which also disables the line. The CPU sending the DLE,EOT need not issue a CONTROL Disable for that line.

When the MODE operand appears along with the TD optype, the function of Control Mode is also performed.

- CONTROL Mode (TM): Used to set the checking mode of the adapter.

IBM 2780 DATA TRANSMISSION TERMINAL

BTAM supports binary synchronous communication with the remote IBM 2780 Data Transmission Terminal over both point-to-point and multipoint line connections. The remote 2780 may be connected point-to-point over either a switched or nonswitched network.

GENERAL INFORMATION

There are three available transmission codes for the 2780: 6-Bit TRANSCODE, EBCDIC, and USASCII. Code charts for these codes are shown in Appendix E.

The 2-character data link control functions in the three codes vary in the second character used, as shown in Figure 30.

	Characters used in:		
Function	USASCII	EBCDIC**	TRANSCODE
ACK-0	DLE,0	DLE,X'70'	DLE,-(hyphen)
ACK-1	DLE,1	DLE,X'61'	DLE,T
WABT*	DLE,?	DLE,X'7F'	DLE,W

*The WABT function is not used by the 2780 although the character representation is defined as shown in the three codes.

**The second characters in the EBCDIC code are shown in their hexadecimal representation.

Figure 30. Data Link Control Functions in USASCII, EBCDIC, and TRANSCODE

In certain cases (e.g., polling/addressing sequences) the user is required to define the hexadecimal equivalents of the transmission code characters in main storage. If the transmission code is EBCDIC there is, of course, no problem. For USASCII and TRANSCODE the user must remember that each 7-bit or 6-bit character, respectively, is to be represented hexadecimally within an 8-bit System/360 byte, all eight bits of which are sent over the System/360 channel to the control unit on a write operation. The following rules hold:

- In main storage, bits 1-7 in a System/360 byte correspond to bits b_7-b_1, respectively, of the USASCII character. The zero-bit in the System/360 byte is always zero (OFF). When the control unit receives a byte over the channel, a parity bit is sent over the line along with bits 1-7 of the byte. Conversely, when 7 bits plus a parity bit are received by the control unit from the line, the 7 (data) bits are read into main storage right-justified in a byte and the zero-bit is set to zero. Example: The hexadecimal equivalent (in main storage) of the USASCII character, A, is 41.

- For TRANSCODE, a similar rule holds. The hexadecimal equivalent is right-justified in a System/360 byte (bits 2-7) and the zero-bit and 1-bit are always set to zero (OFF). Only bits 2-7 are sent over the line. Example: The hexadecimal equivalent of the TRANSCODE character, A, is 01.

For the 2780 there are 4 characters in each code that are not data link control characters but that provide secondary end-to-end control functions. These are:

1. BEL - The Bell character is used only

in 2780-to-2780 communication and not in CPU-to-2780 communication. Its function in the terminal-to-terminal environment is to cause the audible alarm to sound when it is received by the 2780.

2. EM - The "end of medium" character has a twofold function. It causes the card reader to stop reading a card and it causes a punch eject when received by the punch (in normal text mode for both cases). It does, however, appear on the line or the punched card along with the data characters.

3. ESC - The escape character is the first character in a 2-character sequence providing a dual function:

   - On a point-to-point line the escape sequence is used for component selection (i.e., of the printer or the punch). Component selection is discussed in the section, Point-to-Point Communication. For multipoint lines the techniques of polling and addressing are used and the function of component selection through escape sequences is not applicable.

   - On either point-to-point or multipoint lines the escape sequence is used for vertical forms control of the printer. Vertical forms control is achieved by specifying one of the following 2-character sequences as the first two characters in a block of data (i.e., a print line record) transmitted to the printer.

Sequence		Forms Motion
USASCII	*EBCDIC/TRANSCODE	AFTER Print
ESC,Q	ESC,/ (Slash)	Single Space
ESC,R	ESC,S	Double Space
ESC,S	ESC,T	Triple Space
ESC,A	ESC,A	Skip to 1
ESC,B	ESC,B	Skip to 2
ESC,C	ESC,C	Skip to 3
ESC,D	ESC,D	Skip to 4
ESC,E	ESC,E	Skip to 5
ESC,F	ESC,F	Skip to 6
ESC,G	ESC,G	Skip to 7
ESC,H	ESC,H	Skip to 8

*In EBCDIC the PRE (Prefix) character = ESC

When single spacing is desired and no component selection is required (for a point-to-point connection), the escape sequence (ESC,/ or ESC,Q) may be omitted. For the other vertical forms controls, the appropriate escape sequence must be sent at the beginning of each record, so that the desired forms motion can be accomplished for that record block.

4. HT - The horizontal tab character is used following ESC to indicate the start of a format record. A format record is stored by the 2780 and used to format subsequent blocks received for the printer. HT is used within a format record to indicate where a stop is desired on the print line. A space character in the format record indicates that a stop is not desired at the print line position corresponding to the position occupied by the space character in the format message. No character other than space (SP) or horizontal tab (HT) is permitted in a format message.

In the example of a format message shown in Figure 31 there are no stops defined beyond print position 16. A format message must always be sent in nontransparent mode. A format record is maintained until:

- A new format record is received by the 2780.

- Power is removed from the terminal.

- A card is read by the card reader (EBCDIC and USASCII only).

- A record is sent to the punch (EBCDIC and USASCII only).

For terminals using 6-Bit TRANSCODE, the format record is retained even if card reading or punching occurs.

$$\text{STX}\ \underbrace{\text{ESC HT}}_{1}\ \underbrace{\text{SP SP SP SP}}_{2}\ \underbrace{\text{HT}}_{3}\ \underbrace{\text{SP SP SP SP SP SP}}_{4}\ \underbrace{\text{HT}}_{5}\ \underbrace{\text{SP SP SP}}_{6}\ \underbrace{\text{HT}}_{7}\ \text{ETB}$$

1. Two-character sequence introducing the format record.

2. Indicates print positions 1-4 do not contain a stop.

3. Indicates a stop at print position 5.

4. Indicates no stop in print positions 6-11.

5. Indicates a stop at print position 12.

6. Indicates no stop in print positions 13-15.

7. Indicates a stop at print position 16.

Figure 31. Example of a Format Record

The HT character is also used in subsequent data blocks (once a format record has been stored) to define a skip to the next stop position on the print line. For example, if the user desired to print two characters (A,B) at print positions 1 and 2 and three characters (C,D,E) at print positions 12, 13, and 14, he could send the message (with the format record in Figure 31 stored):

STX A B HT HT C D E ETB

If HT appears in nontransparent data when no format record is stored or if it appears in the data beyond the position defined as the last stop on the print line, an error occurs (overrun of the print line).

Format records can be stored and used by the 2780 only if it is equipped with the Printer Horizontal Format Control feature. Formatting of blocks transmitted to the 2780 in transparent mode is not performed, because the HT character is not recognized as an end-to-end control character. Instead, it would be treated as a data character. If the 2780 is not equipped with the Printer Horizontal Format Control feature, the HT character is not recognized apart from data characters, even in nontransparent mode.

The 2780 has a 400-position line buffer providing service for the data link with the CPU. The basic 2780 has the capability of receiving and sending two records per transmission (without line turnaround until after the second record) by use of intermediate block checking mode. In this mode the first record is ended by the unit separator (US) character (in all codes) which provides the ITB function, while the second record is ended by ETB or ETX. Thus, in nontransparent mode, messages appear on the line in the format:

STX---Text---US---Text---ETB(or ETX).

(The block check characters are not shown.)

In transmission to the 2780 only the "text" is printed or punched. Note that US, ETB, or ETX provides the function of punch eject without being punched themselves (unlike EM).

The STX (Start-of-Text) character is mandatory at the beginning of the first record and optional at the beginning of the second (after US). When a 2-record message is received by the 2780 from the CPU, each record is individually checked. If the first intermediate block is valid and the second is invalid, the first record is printed or punched, and the second is not. A subsequent retransmission of the 2-record message results in the printing or punching of the second record, only, if it is valid. If the first record is invalid, neither record is printed or punched, to avoid processing out of sequence.

In transparent mode (may be used only with the EBCDIC transmission code) BTAM supports only single-record transmission to the 2780. The format of such a message is:

DLE, STX---Text---DLE, ETB (or DLE, ETX).

However, the 2780 transmits (from the card reader) transparent messages of the form:

DLE, STX---Text---DLE, US DLE,
STX---Text---DLE, ETB (or DLE, ETX)

The distinction between ETB and ETX is that ETX is used to end the last block in a transmission. Transparent mode is not used in communication with the 2780 when the transmission code is USASCII or 6-bit TRANSCODE.

	Number of Records per transmission	Number of data characters[1] per record
Nontransparent Mode	Maximum: 2 without MRT[2] 7 with MRT	<u>Variable record length</u> Maximum: 80 (to punch) 80 (from reader) print line[3] (to printer)
Transparent Mode	Maximum: 1 (to 2780) 2 (from 2780 without MRT) 4 (from 2780 with MRT)	<u>Fixed record length</u> 80 (to punch) print line (to printer) 80(from reader)

[1]Not including control characters (STX, US, etc) or escape sequences.

[2]MRT=Multiple Record Transmission feature in 2780.

[3]The print line may be 80, 120 or 144 positions.

Figure 32. Block Number and Block Length Considerations

If the 2780 is equipped with the Multiple Record Transmission feature, up to seven records may be sent or received by the 2780 in nontransparent mode. In transparent mode (EBCDIC) the 2780 sends up to four records per transmission. Only one record per transmission may be sent to the 2780 using BTAM support in transparent mode. Figure 32 summarizes block number and block length considerations.

Although escape sequences cannot perform the function of component selection (see <u>Point-to-Point Communication</u>) in transparent mode, they may be used in transparent transmission to the printer to perform vertical forms control. They are optional in this case, however, and do not count as part of the fixed record length (80, 120, or 144 depending on the printer used). The only case in which the escape sequence is mandatory in transparent mode is if the first data character in the record coincides with the escape (ESC).

When intermediate block checked messages are received by the 2701 or 2703 control unit, an error information byte (EIB) is inserted following each US, ETB, or ETX and sent over the System/360 channel into main storage. The EIB has the following significance:

- No bits ON = No error in the intermediate block.
- 4-bit ON = Data check in the block.
- 5-bit ON = Overrun.

The user generally need not concern himself with the EIB's because BTAM automatically attempts recovery when an error occurs. A completion code in the DECB informs the user when an irrecoverable error occurs. The completion code fully defines the error. The user needs to consider the presence of EIB's in applications involving message switching, in which he would not want to send the EIB's to some other terminal. To avoid reception of an EIB by the remote terminal, change the EIB to a SYN character before transmitting it. The remote binary synchronous control unit automatically strips out the SYN character when it is received.

Messages from the 2780 are always sent in intermediate block check mode. For this reason it is essential that the problem program set the 2701 or 2703 in intermediate block check mode via the MODELST operand of the DTFBT macro or via the CONTROL macro (see that section) prior to communication with the 2780.

When a message is received in transparent mode, the 2701 (or 2703) strips out the DLE character preceding US, ETB, or ETX and does not send it over the channel into main storage. The EIB, of course, is inserted following the US, ETB, or ETX, as in nontransparent mode. In all cases the EIB character is sent into main storage under the same (CCW) count control as any other character.

Programming Notes

The following considerations are important to the programmer:

- In nontransparent mode, maximum record length is 169 data characters. Records exceeding 169 data characters cause an I/O buffer overrun error. The latter error results in the 2780 sending an EOT reply to the message. The 2780 would then be in control mode.

  Records exceeding 80 characters (for the punch) or the print line (for the printer) are truncated without error when received by the 2780 as long as they do not equal or exceed 170 characters.

- In nontransparent mode, variable-length records may be sent to the 2780. In transmitting to a 2780 equipped with the multiple record transmission feature, the programmer can send up to 7 records in a transmission, provided the number of characters does not exceed the 400 positions in the line buffer. The STX character preceding a record is stripped out by the receiving 2780 control unit and does not reach the line buffer. All other characters, including escape sequences, end-to-end controls, and US, ETB, or ETX occupy positions in the buffer.

- If blank cards (for subsequent punching) are in the 2780 card reader hopper behind a deck being transmitted to the CPU, the first blank card will cause the 2780 (if equipped with the Auto Turnaround Feature) to stop reading and send an ETX over the line. The blanks are not sent over the line. In this case the last message from the 2780 will be:

  STX---Text---US ETX or
  STX ETX.

- Nontransparent and transparent modes cannot occur in the same transmission to the 2780. That is, a message of the form:

  STX---Text---US, DLE,
  STX---Text---DLE, ETB

  is not permitted.

- There are two responses by the 2780 to abnormal conditions occurring while the 2780 is receiving a message from the CPU: NAK and EOT.

  NAK is used for the class of errors which can result from line errors. For example, the block check character(s) (inserted by the 2701 or 2703 control unit) following a message block transmitted to the 2780 can be invalid, indicating to the 2780 that the received message block was in error. A NAK response by the 2780 requests a retransmission. Another type of line error can result in a message block received by the 2780 appearing to contain too many records or too many characters (although a problem program error can produce the same result). A NAK response by the 2780 requests retransmission to correct the line error. BTAM ERP automatically attempts recovery (see the RETRY operand of DTFBT) in these cases. Of course, if the problem program has actually transmitted too many records or characters within a message block, it involves a correction of the problem program for successful recovery.

  EOT is the response used by the 2780 for detectable errors in the problem program (i.e., using horizontal format control without first storing a format record, transmitting a single record having too many characters in a message block, etc.) and also for the occurrence of 2780 internal errors. The latter errors can be the result of punch jam, printer forms check, buffer parity check, etc. When they occur, the 2780 generally responds immediately with an EOT, thereby informing the transmitting CPU of a condition at the 2780 requiring the intervention of the operator before normal operation can be resumed.

  Recovery from such a disruption in the normal operation is dependent on the system discipline of the particular application. That is, both the problem program restart and recovery procedures and the instruction guide for the 2780 operator must be compatible. The operator can be instructed to prepare the 2780 to expect the message to which EOT was responded to be retransmitted, or alternatively, the operator can prepare it to receive a new message unrelated to the previous message. The problem program should adhere to the same discipline.

  In certain instances the 2780 will not immediately respond with EOT when an internal error occurs. This occurs only if the error condition arises during the printing or punching of the last record in a received message block. The 2780, upon verifying that the last record in a received block is free of error, responds with the appropriate acknowledgement to the CPU and overlaps the processing of the last record with subsequent transmission. If an internal error occurs during the processing of the last record, the 2780 responds EOT to the subsequent transmission. The operator can intervene to process the record on which the error actually occurred without retransmission of that record. Thus, restart and recovery procedures by the problem program can begin by retransmitting the subsequent transmission.

A description of internal errors and operator procedures may be found in the SRL publication on the 2780 hardware.

When the 2780 is transmitting to the CPU, special use is made of the ENQ character to denote the occurrence of a 2780 internal error during the transmission. If the 2780 transmits a message of the form:

STX---Text--US--Text--ENQ or
STX--┘Text--ENQ

it indicates that a buffer parity or overrun error occurred in the message being transmitted by the 2780. Such a message is invalid and BTAM ERP automatically responds with an NAK (negative acknowledgment) to the 2780. The 2780, in turn, transmits the EOT character to the CPU indicating that transmission is aborted until operator intervention corrects the error condition.

If the 2780 transmits a message of the form:

STX ENQ (no text)

it indicates that an internal error occurred in the processing of a record being readied for transmission to the CPU. The exchange of NAK (CPU-to 2780) and EOT (2780-to-CPU) proceeds as discussed in the preceding paragraph.

BTAM posts completion code 41 and sets bit 6 of the error information field and bit 1 of the flag byte, in the DECB to inform the user of the error condition and the reception of EOT. Subsequent transmission by the 2780, after contact is again resumed between the CPU and the 2780, normally begins with the record on which the error occurred.

POINT-TO-POINT COMMUNICATION

Point-to-point communication with the 2780 may occur over switched or nonswitched networks. In point-to-point communication, the programmer must perform component selection when transmitting messages to the 2780.*

Component selection is accomplished by means of a 2-character escape sequence appearing at the beginning of a nontransparent record. To select the punch, ESC, 4 is sent; to select the printer, any one of the vertical forms control escape sequences may be sent (e.g., ESC /, ESC A, ESC B, etc). Thus, the vertical forms control escape sequences perform the dual function of selecting the printer and providing control of forms motion for the record blocks in which they appear. Once a component is selected it need not be reselected in successive record blocks intended for it. Because component selection occurs only in nontransparent mode, the first message in a transmission to the 2780 cannot be in transparent mode. Figure 33 illustrates component selection.

*That is, if the Mode Switch on the 2780 is on Transmit or Receive, component selection is mandatory. If the Mode Switch is on Print or Punch, component selection is ignored and all messages received by the 2780 are automatically printed or punched. The Mode Switch on the 2780 is manually set by the attending operator.

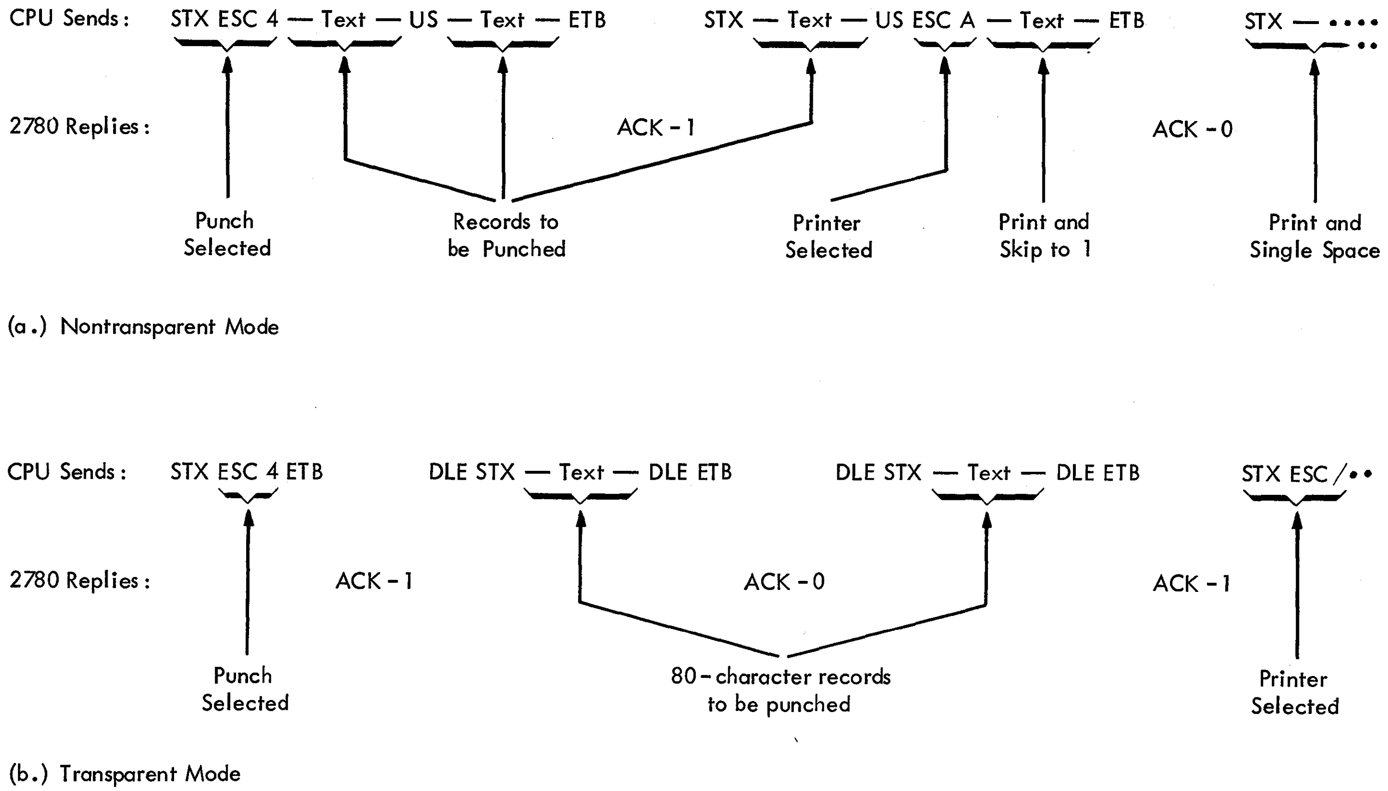


Figure 33. Examples of Component Selection

Contention

In a contention system, either the CPU or the 2780 may initiate transmission by sending the inquiry signal (ENQ) over the line. When a true contention situation occurs (both the CPU and the 2780 are bidding for control of the line), the 2780 must be declared the master station. For this reason, the programmer must code SLV in the FEATURE operand sublist in the DTFBT for the line. Thus, when an ENQ is received by the CPU in reply to an ENQ transmitted to the 2780 via a WRITE initial, BTAM posts completion code 50 in the ECB and returns control to the user. By then issuing a READ initial, the user can receive from the remote 2780.

If the CPU successfully gains control of the line (ACK-0 received as reply to ENQ), the first message with the component selection sequence is sent to the 2780. Figure 33 illustrates typical transmission sequences.

When the 2780 is the transmitting station, it sends ETX (or DLE,ETX) at the end of the last record block transmitted. Upon receiving a positive reply it transmits EOT.

When the 2780 is equipped with the auto turnaround feature, the card punch automatically obtains ready status after the card reader reads a blank card. This permits the CPU (upon receiving EOT) to bid immediately for the line and select the punch for a punching operation. Without this feature, selection of the card punch following a card reader operation requires operator intervention. Operator intervention is not required for the printer to be selected following a card reader operation.

Figure 34 lists the READ/WRITE options available for communication with the 2780 in a contention system. A detailed explanation of the channel programs may be found in the section CPU-to CPU Contention.

Option	Type Code	Point-to-Point		Multipoint
		Contention	Dial	
READ Initial	TI	X	X	X
READ Initial Inquiry	TIQ	X		
READ Continue	TT	X	X	X
READ Repeat	TP	X	X	X
READ Inquiry	TQ	X	X	X
WRITE Initial	TI	X	X	X
*WRITE Initial Transparent Block	TIE			X
*WRITE Initial Transparent Text	TIX			X
WRITE Continue	TT	X	X	X
*WRITE Transparent Block	TE	X	X	X
*WRITE Transparent Text	TX	X	X	X
WRITE End-of-Transmission	TR	X	X	X
WRITE Inquiry	TQ	X	X	X
WRITE Disconnect	TD		X	
CONTROL Disable	TD		X	
CONTROL Mode	TM	X	X	X

*Available for EBCDIC transmission code only.

Figure 34. READ/WRITE/CONTROL Options for the 2780

Dial

Transmission over a switched network is initiated when either the CPU or the operator at the remote 2780 performs a calling operation. For this reason both answering (via READ initial) and calling via READ initial or WRITE initial) operations are available for communication with the 2780 over a switched network. ID verification is not allowed in either case. Thus the READ initial for communication with the 2780 is identical to that described in the section CPU-to-CPU Dial for no ID verification.

For the calling operation the user must supply, via the entry operand in READ initial or WRITE initial the address of a DIALST type terminal list. For this reason the problem program must include a DFTRMLST macro instruction as shown below.

Name	Operation	Operands
symbol	DFTRMLST	DIALST, dialcount-integer, dialchars-decchars

The dialcount operand indicates the number of dial characters to be used, while the dialchars operand consists of the decimal digits in the phone number to be dialed. Thus, if the phone number of the remote 2780 is 123-4567, the user would code:

```
MYLIST DFTRMLST DIALST,7,1234567
```

The 2780 normally does not transmit the disconnect signal (DLE, EOT) to the CPU. This permits the problem program to determine when the connection is to be broken. Thus, when the 2780 is the transmitting station, it sends EOT after receiving a positive reply to its last message to be transmitted. The problem program may then send the disconnect signal or bid for the line (via WRITE inquiry) to subsequently transmit to the 2780. When the problem program sends EOT on a dial line, the 2780 reacts in either of two ways:

- If the card reader is ready to send, the 2780 bids for the line by sending ENQ to the CPU. BTAM posts completion code of 7F for this case. The user may then follow the completed WRITE TR by a READ continue (TT) if the program is ready to receive.

- If the card reader is not ready to send, the 2780 will not respond to the EOT. Instead, the 2780 continues to monitor the line after receiving the EOT. After a nominal twenty seconds, the 2780 (with the Auto Answer feature) will go "on-hook" if no bid (ENQ) is received from the CPU.

Once a line connection is established between the CPU and a 2780 over a switched network, message transmission proceeds in the same fashion as described for a contention system. Component selection and auto turnaround operate the same in both systems. The READ/WRITE/CONTROL options are listed in Figure 34 for a dial operation. A complete description of the channel programs may be found in the section, CPU-to-CPU Dial.

MULTIPOINT COMMUNICATION

Transmission of messages from or to a particular 2780 on a multipoint line is initiated by the CPU, which places the line in control mode and then sends the appropriate polling or addressing sequence.

2780 to CPU

Message transmission from a remote 2780 to the CPU is initiated by the CPU through a READ initial macro instruction. Execution of the READ initial channel program results in the sending of a polling sequence by the CPU and a subsequent response by the polled 2780. Polling is accomplished using Auto Poll -automatic negative response polling. A negative response to polling does not cause a CPU interruption until:

1. The last entry in the AUTOLST type terminal list is polled.

2. A RESETPL macro is issued by the problem program and the last entry in the AUTOWLST type terminal list is polled.

Name	Operation	Operands
symbol	DFTRMLST	listype-{AUTOLST }, {AUTOWLST} eotsyn-absexp, {pollchars-hexchars,}...

In issuing the DFTRMLST macro in the form shown, the user is required to specify, via eotsyn, the bit configurations of the EOT and SYN characters in the transmission code used. These are:

Code	Bit Configuration of EOT,SYN (in hex)
EBCDIC	X'3732'
USASCII	X'0416'
TRANSCODE	X'1E3A'

For the polling sequence, the user specifies the hexadecimal equivalent of a 3-character sequence of the form:

x 6 ENQ

where the x may be any uppercase or lowercase character used as the station address. This is set at the time of installation of the 2780. The second character is always a 6 to indicate the card reader. Up to 98 entries may be specified in the terminal list. The list generated by BTAM is shown in Appendix A.

READ Macro Instruction

For multipoint operation there are four options available for the READ macro instruction. The optypes used are (see also Figure 34):

TI - READ Initial

TT - READ Continue

TP - READ Repeat

TQ - READ Inquiry

The latter three options are the same as that described in the section CPU-to-CPU Contention.

READ Initial (TI): Depending upon whether the terminal list used is the AUTOLST or AUTOWLST type, one of two separate channel programs is generated and executed. When the terminal list specified via the entry operand is the AUTOLST type, the channel program is:

(1)	Send EOT, SYN, SYN, SYN	Note 1
(2)	Poll using the AUTOLST	Note 2
(3)	I/O No-Op	Note 3
(4)	Read index byte and message	Note 4

Note 1: Send EOT to place the line in control mode.

Note 2: Start the Auto Poll of the list with the entry specified in the READ macro.

Note 3: If the Poll command is nonproductive (all polled terminals have responded negatively) the Poll command ends and chains to the I/O No-Op, causing a CPU interruption. (Negative responses are never sent into main storage.)

Note 4: The Poll ends and chains to the Read command on any non-negative response to polling (including such cases as no response - i.e., timeout). In this case the first character read into the user's input area is the index byte (see Appendix A) of the entry last polled. This is true even for a terminal that was polled and did not respond at all. The second character in the input area is the first character (if any) of the received message.

When the terminal list specified via the entry operand in READ initial is the AUTOWLST type, the channel program is:

(1)	Send EOT, SYN, SYN, SYN	Note 1
(2)	Poll using the AUTOWLST	Note 2
(3)	TIC to (5)	Note 3
(4)	Read index byte and message	Note 4
(5)	Poll from beginning of AUTOWLST	Note 5
(6)	TIC to (5)	Note 6
(7)	Read index byte and message	Note 7

Note 1: Send EOT to place the line in control mode.

Note 2: Start the Auto Poll of the list with the entry specified in the READ macro.

Note 3: After the last entry in the list has been nonproductively polled (EOT response received), the Poll command ends and chains to the Transfer-in-Channel command, which causes command (5) to be executed.

Note 4: On any non-negative response, the Poll command (2) chains to the Read. The first character read into the user's input area is the index byte (See Appendix A) of the entry last polled. The second character in the user's input area is the first character of the received message (i.e., STX). When the message has been successfully read into main storage, the macro is posted complete by BTAM.

Note 5: The Auto Poll of the list continues with the first (active) entry in the terminal list.

Note 6: After the last entry in the list has been nonproductively polled, the Poll chains to the TIC causing the Auto Poll to start again at the beginning of the list.

Note 7: As explained in Note 4, on a non-negative response the Poll chains to the Read command to receive the message into the user's input area (with the index byte).

In issuing the READ initial, the user may specify (via the entry operand) that the Auto Poll start with any entry in the list. The symbol in the name field of the DFTRMLST macro used to generate the polling list is the symbolic address of the first entry in the list and not of the 3-byte header preceding the list (See appendix A for the format of the AUTOLST/AUTOWLST terminal list). In computing the address of an entry in an AUTOLST/ AUTOWLST list, the user should realize that each entry consists of the number of characters in the polling sequence (including ENQ) plus the index byte plus the SYN byte. This means that the fourth entry in the list has a displacement of 12 from the beginning of the list.

A polling operation is illustrated in Figure 35.

CPU to 2780

Message transmission from the CPU to a multidropped 2780 is initiated by the CPU through an initial-type WRITE (TI,TIE, TIX) macro instruction. Execution of one of these WRITE's results in the sending of an addressing (selection) sequence and subsequent transmission of the first message upon receiving a positive response to selection from the appropriate 2780. Component selection in the form used for point-to-point communication is not operative on a multipoint line. Therefore a component (printer or punch) must be selected, via an addressing sequence, prior to any transmission of messages to it. Vertical forms control, however, does function precisely in the same fashion on both point-to-point and multipoint connections.

The addressing sequences are specified by the user through a DFTRMLST macro instruction. The address of the first character of the addressing sequence is specified by the entry operand of the WRITE macro.

Name	Operation	Operands
symbol	DFTRMLST	OPENLST, ({addrchars-hexchars,} ...)

An addressing sequence for the 2780 consists of a 2-character sequence followed by ENQ, where:

x 3 ENQ = Addressing sequence for the printer

x 4 ENQ = Addressing sequence for the punch

Again, the x may be any uppercase or lowercase character corresponding to the station address set at installation time.

The DOS Assembler restriction of 127 characters maximum in a sublist limits the number of addrchars operands coded in a single DFTRMLST macro to a maximum of eighteen (because 2 hexadecimal digits are required to represent a single addressing character).

WRITE Macro Instruction

For multipoint operation, there are eight options available for the WRITE macro instruction. The optypes used are (see also Figure 34):

TI - WRITE Initial

TIE - WRITE Initial Transparent Block

TIX - WRITE Initial Transparent Text

TT - WRITE Continue

TE - WRITE Transparent Block

TX - WRITE Transparent Text

TR - WRITE End-of-Transmission

TQ - WRITE Inquiry

The first three are used to establish initial selection. The latter five provide the same functions as described for these options in the section CPU-to-CPU Contention.

WRITE Initial (TI): This macro instruction is used to establish initial selection of the appropriate 2780 multidropped on the line. The channel program generated and executed is:

(1)	Write EOT, SYN, SYN, SYN	Note 1
(2)	Write user-specified entry	Note 2
(3)	Read response to selection	Note 3
(4)	Write message	Note 4
(5)	Read response to message	Note 5

Note 1: Send EOT to place the line in control mode.

Note 2: Send the addressing sequence specified in the entry operand to select the appropriate multidropped 2780.

Note 3: Read the response to selection. If ACK-0 is received, start command (4). Otherwise, after the user-specified number of unsuccessful retries (see the RETRY operand of DTFBT), terminate the channel program and post completion of the macro (see Completion Codes).

Note 4: Send the message to the 2780.

Note 5: Read the response from the 2780.

WRITE Initial Transparent Block (TIE) and WRITE Initial Transparent Text (TIX): These two macro instructions are identical with the exception of the end characters sent by BTAM at the end of the user's message.

The channel program generated and executed is:

(1)	Write EOT, SYN, SYN, SYN	Note 1
(2)	Write user-specified entry	Note 2
(3)	Read response to selection	Note 3
(4)	Write the user's message	Note 4
(5)	Write the end characters	Note 5
(6)	Read response to the message	Note 6

Note 1: Same as Note 1 for WRITE initial.

Note 2: Same as Note 2 for WRITE initial.

Note 3: Same as Note 3 for WRITE initial.

Note 4: Send the user's message (terminated by count) to the 2780.

Note 5: Send the BTAM-provided end characters:
- If optype TIE is specified, send DLE, ETB.
- If optype TIX is specified, send DLE, ETX.

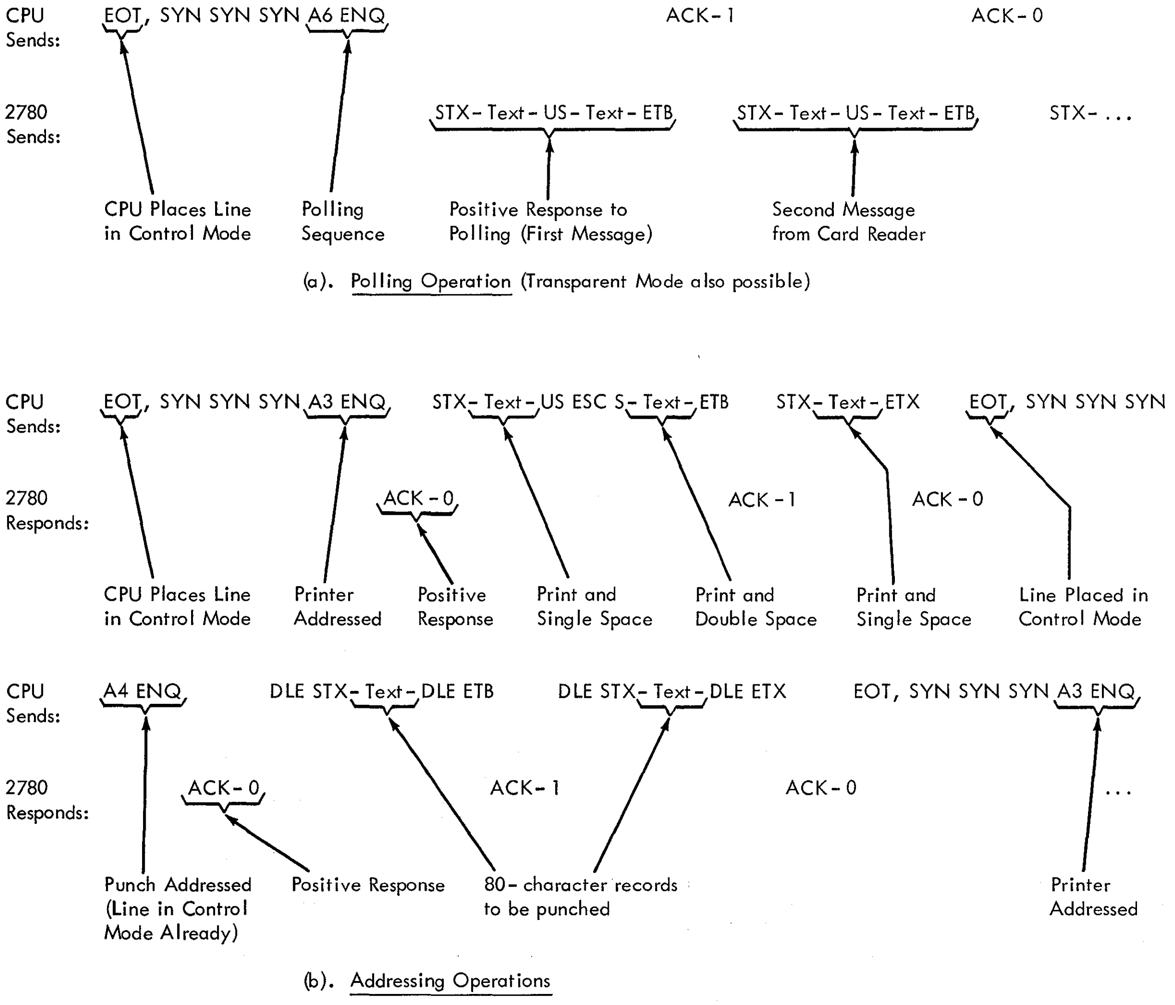
Note 6: Read the response from the 2780 to the message.

Addressing operations are illustrated in Figure 35.

USE OF THE CONTROL MACRO INSTRUCTION

With the exception that intermediate block checking mode should always be set for communication with the 2780, use of the CONTROL macro is as follows.

- The use of CONTROL for 2780 point-to-point contention is the same as discussed for CPU-to-CPU contention.
- The use of CONTROL for 2780 point-to-point dial is the same as discussed for CPU-to-CPU dial.
- The use of CONTROL for the 2780 on a multipoint line is the same as discussed for CPU-to-CPU Contention.



(a). Polling Operation (Transparent Mode also possible)

(b). Addressing Operations

Figure 35. Polling/Addressing Operations

IBM 1130 COMPUTING SYSTEM

BTAM supports binary synchronous communication with the remote IBM 1130 Computing System for both point-to-point and multipoint line connections. The remote 1130 may be connected point-to-point over either a switched or nonswitched network.

POINT-TO-POINT COMMUNICATION

The BTAM support for communication with the 1130 on a point-to-point line is a compatible subset of the BTAM support for CPU-to-CPU communication. Communication with the 1130 does not permit the following:

1. ID Verification (Dial lines).
2. Intermediate block checking mode.
3. Leading graphics.
4. Conversational communication.
5. WABT (Wait-before-transmitting function).
6. USASCII transmission code (EBCDIC must be used).

Figure 35A (READ/WRITE/CONTROL Options for 1130) lists the options available for communication with the 1130 over both nonswitched and switched networks. The channel programs generated and executed are identical to those described in the sections, CPU-to-CPU Contention and CPU-to-CPU Dial, for the same options for READ/WRITE/CONTROL.

Option	Type Code	1130 Pt.-to-Pt.		1130 Multipoint
		Contention	Dial	
READ Initial	TI	X	X	X
READ Initial Inquiry	TIQ	X		
READ Continue	TT	X	X	X
READ Repeat	TP	X	X	X
READ Inquiry	TQ	X	X	X
WRITE Initial	TI	X	X	X
WRITE Initial Transparent Block	TIE	X	X	X
WRITE Initial Transparent Text	TIX	X	X	X
WRITE Continue	TT	X	X	X
WRITE Transparent Block	TE	X	X	X
WRITE Transparent Text	TX	X	X	X
WRITE Disconnect	TD		X	
WRITE End-of-Transmission	TR	X	X	X
WRITE Inquiry	TQ	X	X	X
CONTROL Disable	TD		X	
CONTROL Mode	TM	X	X	X

Figure 36. READ/WRITE/CONTROL Options for 1130

MULTIPOINT COMMUNICATION

The CPU initiates transmission of messages from or to a particular 1130 on a multipoint line by placing the line in control mode and then sending the appropriate polling or addressing sequence to the 1130.

1130 to CPU

Message transmission from a remote 1130 to the CPU is initiated by the CPU through a READ initial macro instruction. This results in the sending of a polling sequence by the CPU and a subsequent response by the polled 1130. Polling is accomplished using Auto Poll (automatic negative response polling). No CPU interruption occurs on a negative response to polling until:

1. The last entry in the AUTOLST type terminal list is polled.

2. An RESETPL macro instruction is issued by the problem program and the last entry in the AUTOWLST type terminal list is polled.

Name	Operation	Operands
symbol	DFTRMLST	listype-{AUTOLST AUTOWLST}, eotsyn-absexp, {pollchars-hexchars,}...

The user specifies the polling characters to be used by coding a DFTRMLST macro. The listype specified may be either AUTOLST or AUTOWLST, depending upon whether the list generated is to be an open or a wrap-around polling list, respectively.

The eotsyn operand is coded as an absolute expression representing the EBCDIC bit configuration of the EOT and SYN control characters. For example, X'3732' is the hexadecimal equivalent of the EBCDIC EOT and SYN (37 for EOT and 32 for SYN) characters. Any absexp (hexadecimal, decimal, binary, etc.) coded is interpreted by BTAM to represent two bytes: the first byte representing EOT, the second, SYN. The eotsyn operand is required by BTAM for internal use both in generating the list and for skipping an entry when the CHGNTRY macro is issued.

The particular entries to be generated in the terminal list are specified via one or more (maximum of 98) pollchars operands. Each pollchars operand represents a polling sequence for a specific 1130 multidropped on the line. A polling sequence for the 1130 is defined as one EBCDIC polling character followed by the ENQ control character. Thus in coding the pollchars operand, the user would code 4 hexchars of the form xxyy where xx represents the hexadecimal equivalent of the polling character and yy is always 2D, the hexadecimal equivalent of ENQ in EBCDIC. The polling character used for a particular 1130 depends upon the character assigned by the problem program in that 1130 as the polling character.

Appendix A shows the list generated for a DFTRMLST macro having AUTOLST or AUTOWLST coded. Note that the list has one entry for each pollchars operand coded. In addition to the characters specified in a pollchars operand, a list entry includes an automatically generated index byte and a "SYN" byte. The indexing is zero-origin. Hence, the index byte for the first entry in the list contains an index value of zero; the index byte for the second entry contains an index value of one, and so forth.

The index byte can be used by the problem program to identify which entry in the list corresponds to the device that responds to polling with a message. The first byte of the user-specified input area (READ initial only) will contain the index value of the list entry for the device that sends a message. The received message will appear in the input area immediately following the index. For those applications involving message switching, the user should move the SYN (synchronous idle) character into the byte containing the index value prior to switching a received message to some other remote device. This is particularly important when the user switches a message received into buffers, because the output area address is the same as the input area address in this case. If the user transmits the index byte, an error will occur.

If the polling list is exhausted (end of the list is reached and no wrap-around is to occur) and all polled devices have responded negatively, no index byte will appear in the input area. Instead, completion code 54 identifies this case.

The "SYN" byte is generated for reasons concerning BTAM's implementation of the SKIP action in CHNGTRY for AUTOLST and AUTOWLST.

READ Macro Instructions

Four options are available for the READ macro instruction. The optypes used are:

TI - READ Initial

TT - READ Continue

TP - READ Repeat

TQ - READ Inquiry

The last three options are the same as for the 1130 point-to-point contention support and are described in the CPU-to-CPU Contention section.

READ Initial: Depending upon whether the terminal list used is the AUTOLST or AUTOWLST type, one of two separate channel programs is generated and executed.

For the case when the AUTOLST type terminal list is specified, the channel program is:

(1)	Send EOT, SYN, SYN, SYN	Note 1
(2)	Poll using the AUTOLST	Note 2
(3)	I/O No-Op	Note 3
(4)	Read index byte and message	Note 4

Note 1: The line is placed in control mode using a Poll command to send the BTAM-supplied EOT and SYN characters. This Poll command has the chain-data (CD) flag set to one, causing chaining to command 2 to occur.

Note 2: The Auto Poll operation is begun, starting with the list entry specified by the user via the entry operand of the READ initial macro. The polling proceeds automatically through the list as long as negative response (EOT) is received from the polled station. No I/O interruption occurs during this operation.

Note 3: If the polling operation is nonproductive (all negative responses received), the Poll command (command 2) ends and chains to the I/O No-Op. When the I/O No-Op ends, an I/O interruption occurs and BTAM posts completion of the operation.

Note 4: The Poll command (command 2) ends and chains to the Read command on any non-negative response to Polling. A non-negative response includes the normal case of a positive response (i.e., a message) and abnormal cases such as time out (no response received). The first character read into the user's input area is always the index byte of the entry for the device responding to the Poll with a non-negative reply. The second character in the input area is the first character (if any) of the received message.

For the case when the AUTOWLST type terminal list is specified, the channel program is:

(1)	Send EOT, SYN SYN SYN	Note 1
(2)	Poll using the AUTOWLST	Note 2
(3)	TIC to (5)	Note 3
(4)	Read index byte and message	Note 4
(5)	Poll from beginning of AUTOWLST	Note 5
(6)	TIC to (5)	Note 6
(7)	Read index byte and message	Note 7

Note 1: The line is placed in control mode.

Note 2: The Auto Poll operation is started with the user-specified entry.

Note 3: After the last entry in the list has been nonproductively polled (EOT response received), the Poll command ends and chains to the Transfer-in-Channel command which causes command (5) to be executed.

Note 4: The index byte and message (if any) are read into the input area.

Note 5: The Auto Poll operation is restarted at the beginning of the polling list.

Note 6: A Transfer-in-Channel to command 5 is executed.

Note 7: The index byte and message (if any) are read into the input area.

In issuing the READ initial (for either AUTOLST or AUTOWLST) the user may specify via the entry operand that the Auto Poll operation start with any entry in the list.

Note: The symbol in the Name field of the DFTRMLST used to generate the polling list is the symbolic address of the first entry in the list and not of the 3-byte header preceding the list. (See Appendix A for the format of the AUTOLST/AUTOWLST terminal list.) In computing the address of an entry in the list the user should realize that each entry consists of the number of characters in the polling sequence (including ENQ) plus the index byte plus the SYN byte. This means that the fourth entry in the list has a displacement of 12 from the beginning of the list.

CPU to 1130

Message transmission from the CPU to a remote 1130 is initiated by the CPU through a WRITE initial macro instruction, which causes an addressing sequence to be sent selecting the appropriate 1130. If the selected 1130 responds positively to addressing, the message is transmitted from the CPU to the 1130.

The user specifies the addressing sequences through a DFTRMLST macro instruction.

Name	Operation	Operands
symbol	DFTRMLST	OPENLST, ({addrchars-hexchars,}...)

An addressing sequence for the 1130 consists of one addressing character followed by ENQ. Each addressing sequence is specified through an addrchars operand in the sublist as shown in the format illustration above. Each addrchars is coded by the user as a hexchars sequence of the form xxyy, where xx represents the hexadecimal equivalent of the addressing character and yy is always 2D, the hexadecimal equivalent of the ENQ character.

The DOS Assembler restriction of 127 characters maximum in a sublist limits the number of addrchars operands coded in a DFTRMLST macro to a maximum of 25. The format of an OPENLST type terminal list is shown in Appendix A.

Note: The address of an 1130 is set by the problem program in the 1130.

WRITE Macro Instructions

Eight options are available for the WRITE macro instructions. The optypes used are:

TI - WRITE Initial

TIE - WRITE Initial Transparent Block

TIX - WRITE Initial Transparent Text

TT - WRITE Continue

TE - WRITE Transparent Block

TX - WRITE Transparent Text

TR - WRITE End-of-Transmission

TQ - WRITE Inquiry

The first three are used to establish initial selection. The last five provide the same functions as the same options for 1130 contention (point-to-point) and are described in the CPU-to-CPU Contention section.

WRITE Initial: This macro instruction is used to establish initial selection of the appropriate remote 1130 multidropped on the line. The channel program generated and executed is:

(1) Write EOT, SYN, SYN, SYN Note 1
(2) Write user-specified entry Note 2
(3) Read response to selection Note 3
(4) Write message Note 4
(5) Read response to message Note 5

Note 1: EOT is sent to place the line in control mode.

Note 2: The addressing sequence is sent to select the appropriate 1130. The address of the first character in the addressing sequence is specified by the user via the entry operand.

Note 3: The response to selection is read into the DECB. If ACK-0 is received, the channel program.is restarted at command (4). Otherwise, the channel program is ended after retrying unsuccessfully and BTAM posts completion of the macro.

Note 4: The message is sent to the 1130.

Note 5: The response from the 1130 is read into the DECB.

WRITE Initial Transparent Block and WRITE Initial Transparent Text: These macro instructions are the same except for the end characters sent by BTAM at the conclusion of the user's message.

The channel program generated and executed is:

(1) Write EOT, SYN, SYN, SYN Note 1
(2) Write user-specified entry Note 2
(3) Read response to selection Note 3
(4) Write user's message Note 4
(5) Write end characters Note 5
(6) Read response to message Note 6

Note 1: Same as Note 1 for WRITE initial.

Note 2: Same as Note 2 for WRITE initial.

Note 3: Same as Note 3 for WRITE initial.

Note 4: The user's message is sent to the 1130.

Note 5: If optype TIE is specified, the sequence DLE,ETB is sent to end the message. If optype TIX is coded, the sequence DLE,ETX is sent.

Note 6: The response from the 1130 is read into the DECB response field.

Programming Note: The user must specify the exact count of characters to be transmitted. This count includes the user-provided DLE,STX sequence at the beginning of the message as well as the text characters. The BTAM-supplied end characters are not included in the count specified via the length operand in the WRITE macro.

Use of the CONTROL Macro Instruction

CONTROL mode (TM) may be used for all three configurations (leased, dial, multipoint). If used, however, the MODE operand must not specify intermediate block checking mode because this mode is not supported by the 1130. Generally, the CONTROL mode form is not used for the 1130 because the mode can be specified via the DTFBT operand, MODELST.

CONTROL disable (TD) can be used for point-to-point dial operation only. Again, if the MODE operand is coded, it must not specify intermediate block checking mode.

SYSTEM/360 MODEL 20

BTAM supports binary synchronous communication with the remote System/360 Model 20 for both point-to-point and multipoint line connections. The Model 20 may be connected point-to-point over either a switched or nonswitched network.

There are two available transmission codes for the Model 20: EBCDIC and USASCII. Code charts for these codes are shown in Appendix F.

The 2-character data link control functions in the above codes vary in the second character used, as shown in Appendix F.

Note: The WABT function is not used by the Model 20 although the character representation is defined in the above codes.

POINT-TO-POINT COMMUNICATION

The BTAM support for communication with the System/360 Model 20 on a point-to-point line is identical to the BTAM support for CPU-to-CPU communication, if the Model 20 is considered as another CPU. The functions supported for nonswitched lines are the same as those described in the CPU-to-CPU Contention section, and the functions supported for dial lines are the same as those described in the CPU-to-CPU Dial discussion, except that the WRITE Wait-Before-Transmitting macro is not supported in either case by the Model 20.

Figure 35B (READ/WRITE/CONTROL Options for Model 20) lists the options available for communication with the Model 20 over both switched and nonswitched networks.

Note: If a BTAM problem program receives a transparent block of data from the Model 20, text cannot be transmitted as a response.

MULTIPOINT COMMUNICATION

The CPU initiates transmission of messages from or to a specific Model 20 on a multipoint line by placing the line in control mode and then sending the appropriate polling or addressing sequence to the Model 20. The polling or addressing sequence that is sent to the Model 20 may consist of a maximum of seven characters preceding the ENQ character, but only the first character is used by the Model 20 hardware. Any additional characters in the polling or addressing sequence are used informationally by the Model 20 problem and hence are optional.

Model 20 to CPU

READ Macro Instructions

Seven options are available for the READ macro instruction. The optypes used are:

TI - READ Initial

TT - READ Continue

TTL - READ Continue with Leading Graphics

TP - READ Repeat

TPL - READ Repeat with Leading Graphics

TQ - READ Inquiry

TTS - READ Stop

OPTION	TYPE CODE	CPU- Model 20 Contention	CPU- Model 20 Dial	CPU- Model 20 Multipoint
Read Initial	TI	X	X	X
Read Initial Inquiry	TIQ	X		
Read Continue	TT	X	X	X
Read Continue with Leading Graphics	TTL	X	X	X
Read Repeat	TP	X	X	X
Read Repeat with Leading Graphics	TPL	X	X	X
Read Inquiry	TQ	X	X	X
Write Initial	TI	X	X	X
Write Initial Transparent Block	TIE	X	X	X
Write Initial Transparent Text	TIX	X	X	X
Write Initial Transparent Conversational	TIXV	X	X	X
Write Initial Conversational	TIV	X	X	X
Write Continue	TT	X	X	X
Write Transparent Block	TE	X	X	X
Write Transparent Text	TX	X	X	X
Write Transparent Conversational	TXV	X	X	X
Write Conversational	TV	X	X	X
Write Disconnect	TD		X	
Write End-of-Transmission	TR	X	X	X
Write Inquiry	TQ	X	X	X
Control Disable	TD		X	
Control Mode	TM	X	X	X
Read Stop	TTS	X	X	X

Figure 37. READ/WRITE/CONTROL Options for Model 20.

The last six options are the same as for the Model 20 point-to-point support and are described in the CPU-to-CPU Contention section of binary synchronous communication. The READ Initial option is the same as for the 1130 multipoint support and is described in the 1130 Multipoint Communication section.

CPU to Model 20

WRITE Macro Instructions

Twelve options are available for the WRITE macro instructions. The optypes used are:

TI	- WRITE Initial
TIE	- WRITE Initial Transparent Block
TIX	- WRITE Initial Transparent Text
TT	- WRITE Continue
TE	- WRITE Transparent Block
TX	- WRITE Transparent Text
TIXV	- WRITE Initial Transparent Conversational
TIV	- WRITE Initial Conversational
TXV	- WRITE Transparent Conversational
TV	- WRITE Conversational
TR	- WRITE End-of-Transmission
TQ	- WRITE Inquiry

All options except WRITE Initial Transparent Conversational, WRITE Initial Conversational, WRITE Conversational, and WRITE Transparent Conversational are the same as those described in the 1130 Multipoint Communication section. The WRITE Conversational and WRITE Transparent Conversational options are the same as those described in the CPU-to-CPU Contention section.

WRITE Initial Conversational (TIV): This macro is used when the user desires to select a remote Model 20, send a message to the remote Model 20, and receive a response. The response received may be either a message or the appropriate positive acknowledgment. The channel program is:

(1)	Write EOT, SYN, SYN, SYN	Note 1
(2)	Write selection characters	Note 2
(3)	Read response to selection	Note 3
(4)	Write message	Note 4
(5)	Read response to message	Note 5

Note 1: EOT is sent to place the line in control mode.

Note 2: The addressing sequence is sent to select the appropriate Model 20. The address of the first character in the addressing sequence is specified by the user with the entry operand.

Note 3: The response to selection is read into the response field in the DECB. If the response is ACK-0, the channel program is restarted at command (4). Otherwise, the channel program is ended after retrying unsuccessfully and BTAM posts completion of the macro.

Note 4: Write the message block.

Note 5: Read the response. It may be either an appropriate positive acknowledgment or a message. (See WRITE Initial Conversational in the CPU-to-CPU Contention section for the format of the parameter list pointed to via entry operand.)

WRITE Initial Transparent Conversational (TIXV): This macro is used when the user desires to select a remote Model 20, send a transparent message to the remote Model 20, and receive a response. The response received may be either a message or the appropriate positive acknowledgment. The channel program is:

(1)	Write EOT, SYN, SYN, SYN	Note 1
(2)	Write selection characters	Note 2
(3)	Read response to selection	Note 3
(4)	Write message	Note 4
(5)	Write ending characters	Note 5
(6)	Read response	Note 6

Note 1: Same as Note 1 for Write Initial Conversational.

Note 2: Same as Note 2 for Write Initial Conversational.

Note 3: Same as Note 3 for Write Initial Conversational.

Note 4: Same as Note 4 for Write Initial Conversational

Note 5: Write the ending characters, DLE ETX.

Note 6: Same as Note 5 for Write Initial Conversational.

All of the following terminal-list descriptions include a control byte for each terminal or component entry (Figure 36). The individual bit positions within the byte are used for:

Bit Position	Meaning
0	If on, indicates that the entry is the last in the list.
1	If on, indicates that the entry is to be skipped when polling or addressing. Turned on and off with CHGNTRY macro.
2	If on, indicates that the list is a wrap-around list.
3-7	List entry number. Each entry numbered successively starting with 0. This field limits the maximum number of terminal or component entries for lists that are created by the DFTRMLST macro to 32. This field is not presently used by BTAM but is reserved for later use.

OPENLST Format

Each entry in an open list (polling or addressing) consists of a one (1030) or two (all others) byte field for the polling or addressing characters plus the control byte. Example for the 1050 and 1030 is shown in Figure 37.

WRAPLST Format

Wrap-around polling lists differ from open lists in that the format bit (bit 2 in the control byte) is ON for the last entry in the list. An example is shown in Figure 38.

DIALST Format

Terminal lists for 1050s on switched networks are illustrated in Figure 39.

IDLST Format

Terminal lists for TWX 33/35s are illustrated in Figure 40.

SSALST and SSAWLST Format

Terminal lists for start-stop Auto Poll are listed in Figure 41.

The two-byte field which follows the last entry in the terminal list is an offset used by the polling restart routine to get back to the start of the list.

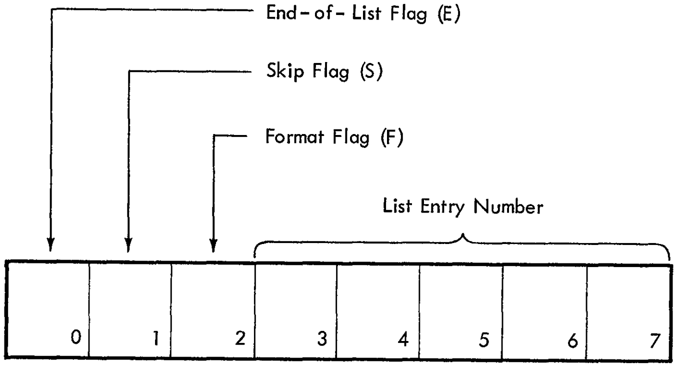
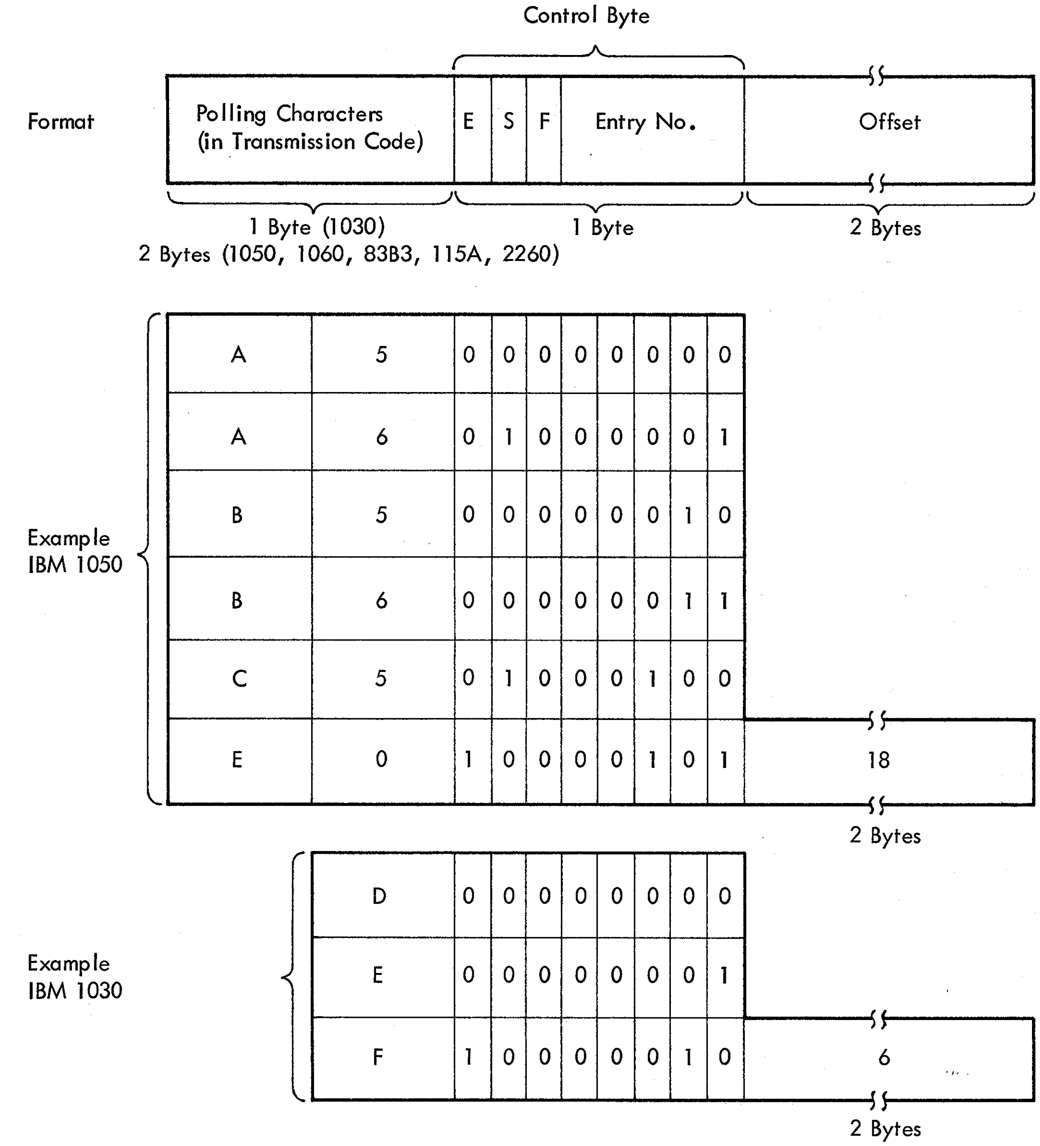


Figure 38. Format of Control Byte

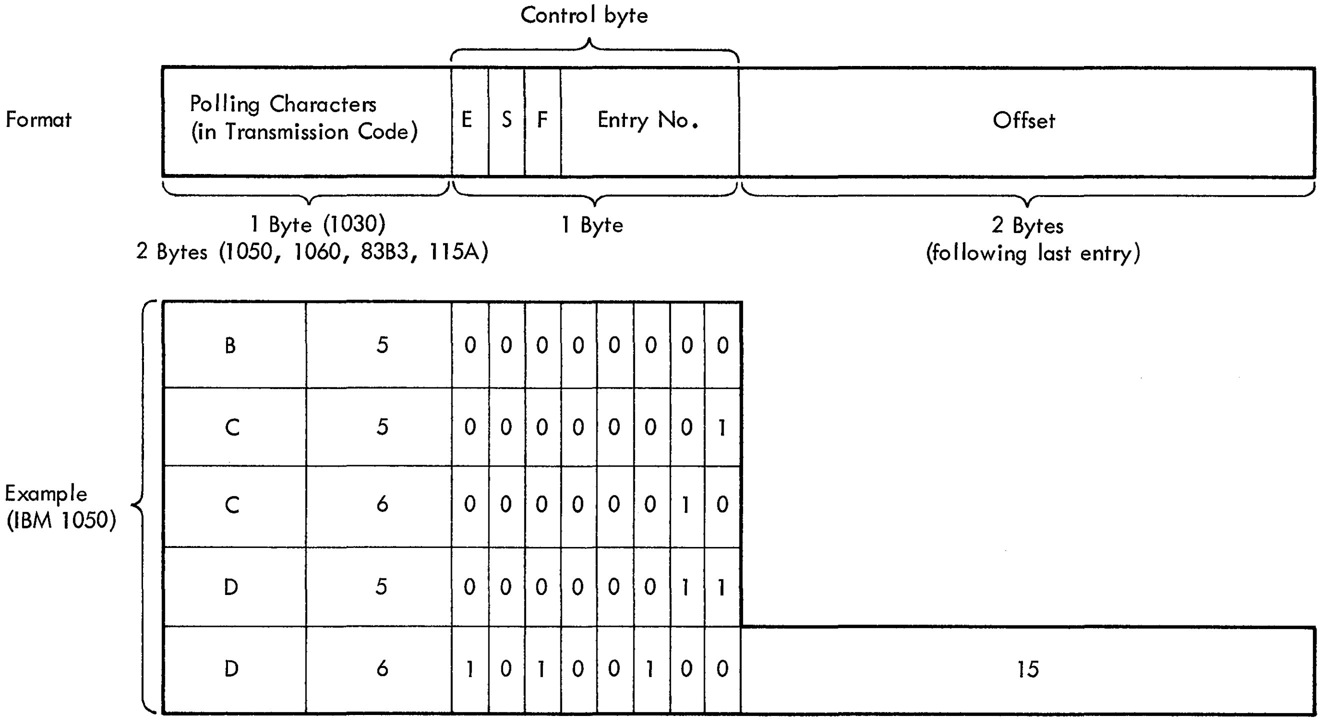


Polling Characters (in Transmission Code)	E	S	F	Entry No.	Offset

A	5	0	0	0	0	0	0	0	0	
A	6	0	1	0	0	0	0	0	1	
B	5	0	0	0	0	0	0	1	0	
B	6	0	0	0	0	0	0	1	1	
C	5	0	1	0	0	0	1	0	0	
E	0	1	0	0	0	0	1	0	1	18

D	0	0	0	0	0	0	0	0	
E	0	0	0	0	0	0	0	1	
F	1	0	0	0	0	0	1	0	6

Figure 39. Open Polling or Addressing List (OPENLST): Format and Examples


Control byte
Format
Polling Characters (in Transmission Code)
E
S
F
Entry No.
Offset
1 Byte (1030)
2 Bytes (1050, 1060, 83B3, 115A)
1 Byte
2 Bytes (following last entry)
Example (IBM 1050)

B	5	0	0	0	0	0	0	0	0	
C	5	0	0	0	0	0	0	0	1	
C	6	0	0	0	0	0	0	1	0	
D	5	0	0	0	0	0	0	1	1	
D	6	1	0	1	0	0	1	0	0	15

Figure 40. Wrap-around Polling List (WRAPLST): Format and Example

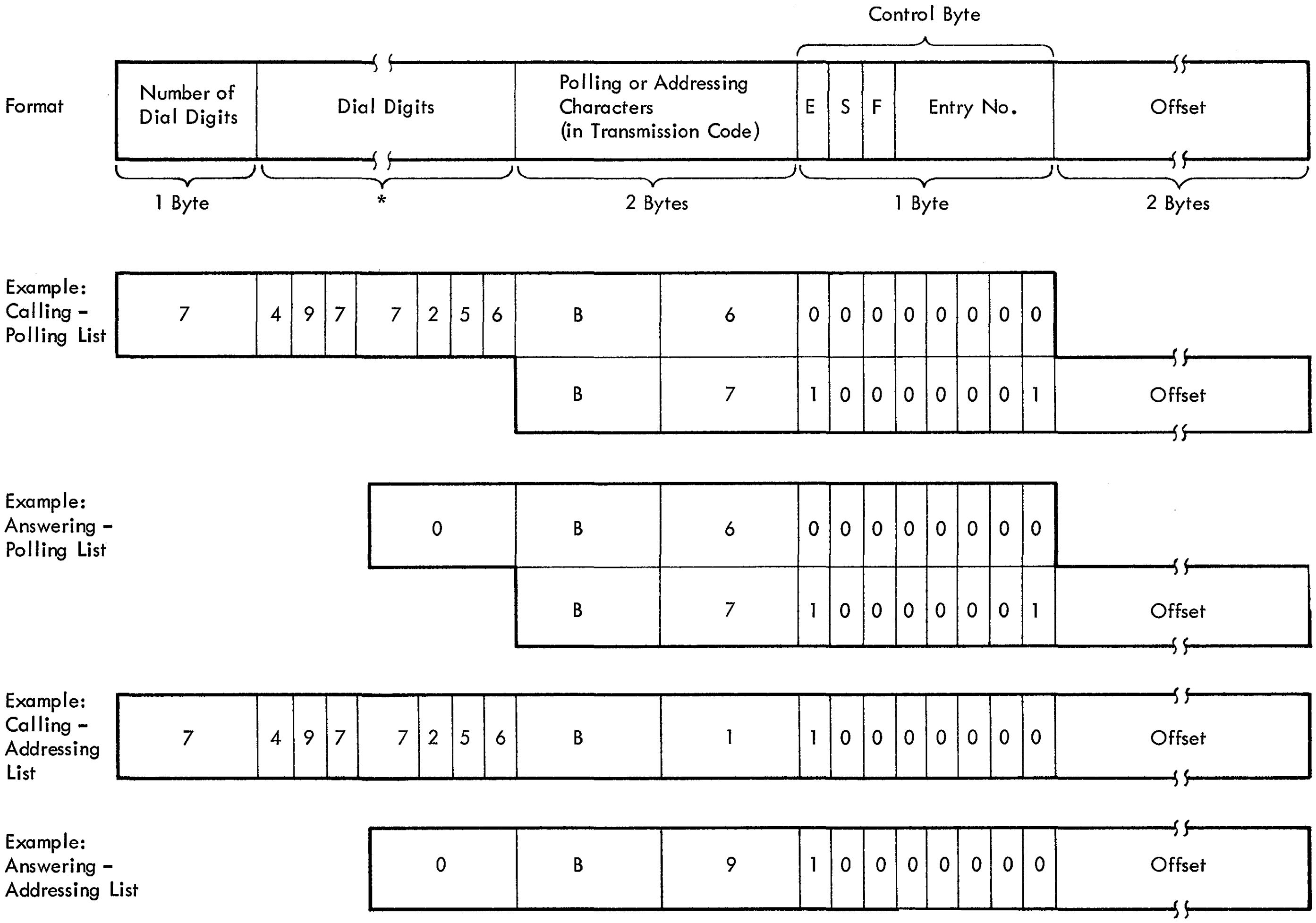
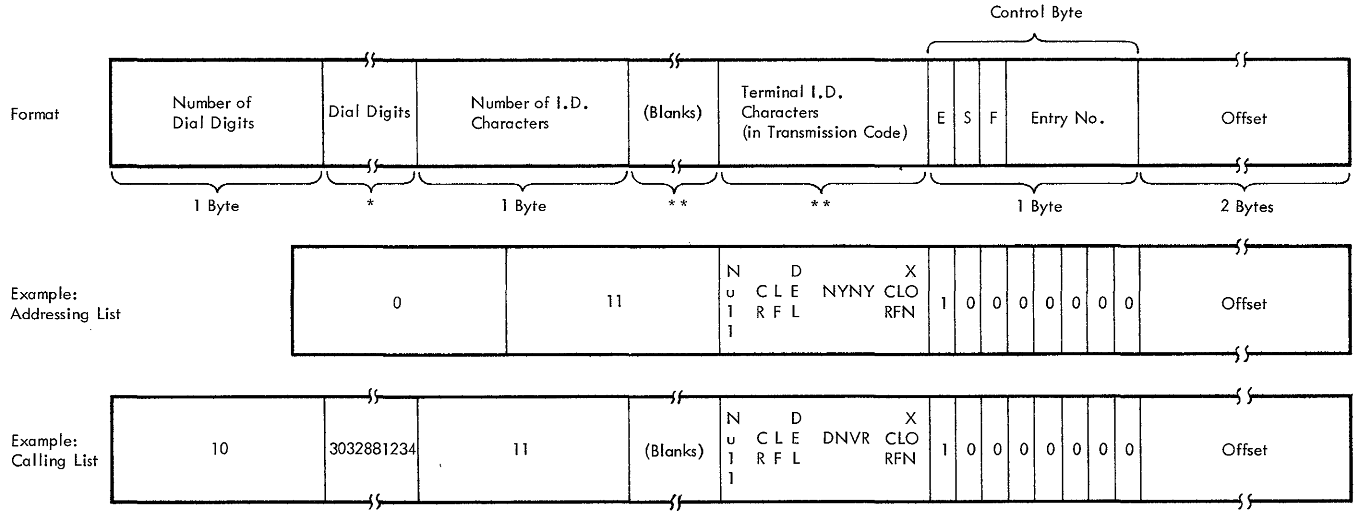


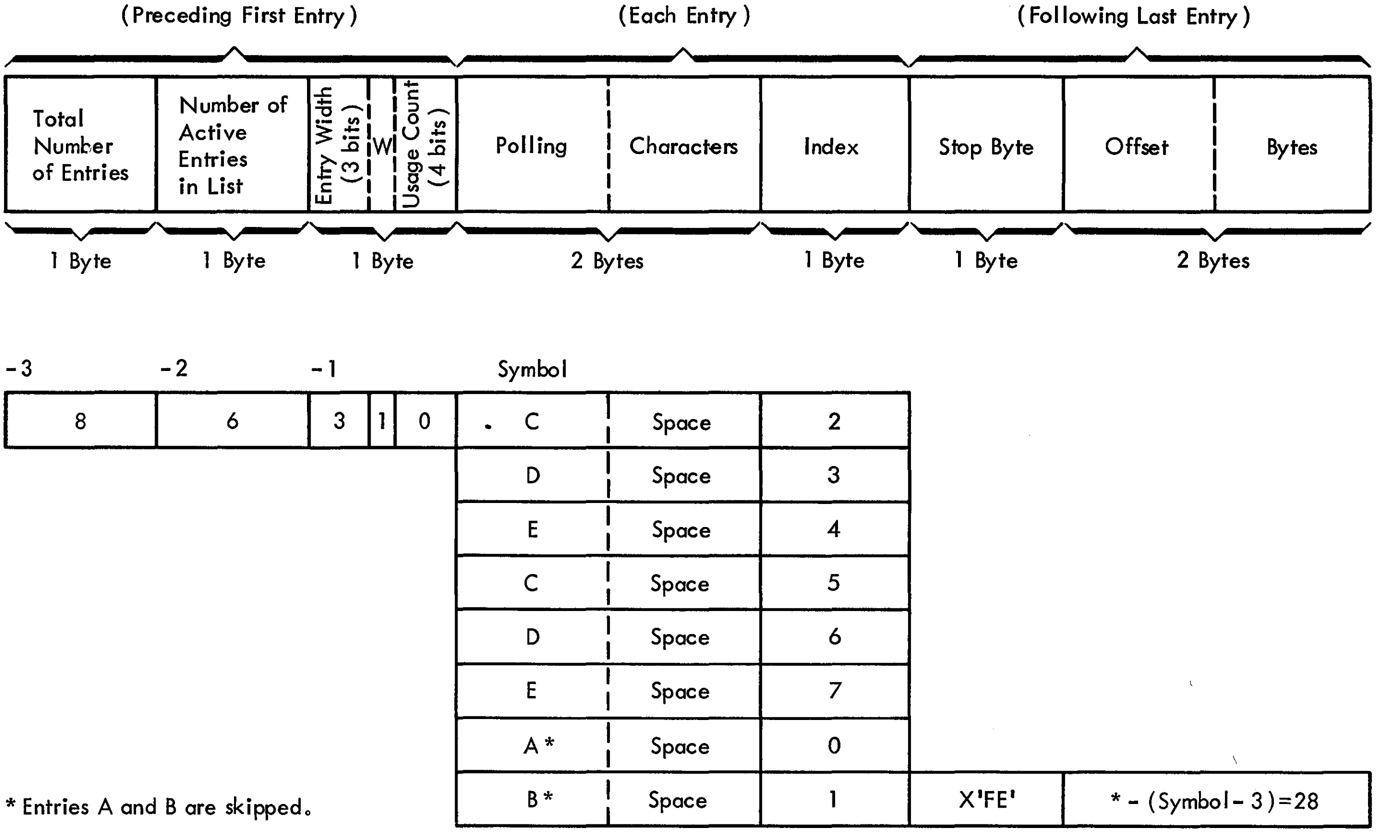
Figure 41. 1050 Terminal List (DIALST): Format and Examples



* Length in bytes equals number of dial digits

** Length in bytes equals number of terminal I.D. characters

Figure 42. TWX-Terminal List (IDLST): Format and Example



-3	-2	-1			Symbol				
8	6	3	1	0	. C	Space	2		
					D	Space	3		
					E	Space	4		
					C	Space	5		
					D	Space	6		
					E	Space	7		
					A*	Space	0		
					B*	Space	1	X'FE'	* - (Symbol - 3)=28

* Entries A and B are skipped.

Figure 43. 2740SC Terminal List (SSAWLST): Format and Example

FORMAT OF DIALST FOR CPU-TO-CPU DIAL

(No ID Verification)

This list is used as a calling list, only.

```
┌────┬──────────────┬───────────┐
│ nd │ dial-chars   │  inlist   │
│    │              │  address  │
└────┴──────────────┴───────────┘
←1d→←─────nd──────→←────3─────→
byte     bytes          bytes
```

n_d: specifies the number (in binary) of dial characters in the list.

dial-chars: specifies the actual characters (in binary) to be used in dialing the remote CPU.

inlist address: used only if the WRITE initial conversational (TIV) or WRITE initial transparent conversational (TIXV) is issued. It specifies the address of the parameter list containing the:

- Address of the input area (to receive the conversation). An address of all zeros means buffering is to be used.

- Length of the input area.

For an answering operation, the DIALST list is a single byte containing all zeros.

FORMAT OF IDLST FOR CPU-TO-CPU DIAL

(For ID Verification)

IDLST (CPU-CPU Dial)

call list:

n_d	dialchars	n_r	0----0	recv-chars	n_s	send-chars	inlist address
1 byte	n_d bytes	1 byte	n_r bytes	n_r bytes	1 byte	n_s bytes	3 bytes

answer list:

0	n_r	0----0	recv-char	n_s	send-chars
1 byte	1 byte	n_r bytes	n_r bytes	1 byte	n_s bytes

n_d: The number (in binary format) of dialing characters to be used in calling the remote CPU.

dialchars: the actual dialing characters (in binary format) used.

n_r: the number (in binary) of identification characters the remote CPU is expected to send. (This byte may be 0.)

0----0: The received ID characters will be read into these bytes.

recv-chars: the (user-supplied) ID characters expected to be received from the remote CPU.

n_s: the number (in binary) of ID characters which the CPU will send to the remote CPU.

send-chars: the actual ID characters which the CPU will send to the remote CPU.

inlist address: the address of a user-maintained parameter list to be used if the CPU is calling and wishes to issue a conversation-type WRITE Initial. The parameter list pointed to must contain 2 fullwords:

Word 1 - address of the input I/O area. (All zeros means buffering is to be used.)

Word 2 - length of the expected message.

Note: Both recv-chars and send-chars are in transmission code representation.

FORMAT OF AUTO POLL TERMINAL LIST (AUTOLST, AUTOWLST)

```
┌─────────────┬────────┬─//─┬───────┬───────────────┐
│3-byte header│entry 1│    │entry n│5-byte trailer│
└─────────────┴────────┴─//─┴───────┴───────────────┘
```

3-byte header:

```
┌───────────┬────────────┬─┬───────────┐
│total count│active count│W│entry width│
└───────────┴────────────┴─┴───────────┘
←─1 byte──→←─1 byte─────→←─1 byte─────→
```

total count: number of entries in the list.

active count: number of active entries in the list.

W (Wrap bit): ON if AUTOWLST; OFF if AUTOLST.

entry width: width in bytes of each entry.

entry (active):

```
┌───────────┬───┬──────┬───────┐
│           │ENQ│Index │SYN    │
└───────────┴───┴──────┴───────┘
←Number of ──────→←─1────→←─1───→
  bytes            byte    byte
  specified
  by user
```

Index: position of entry in list (0-97).

entry (inactive):

```
┌───────────┬───┬──────┬───────┐
│           │EOT│ SYN  │SYN    │
└───────────┴───┴──────┴───────┘
```

5-byte trailer:

```
┌────┬────┬────┬────────────┐
│EOT │SYN │ *  │Offset Count│
└────┴────┴────┴────────────┘
←─1─→←─1─→←─1─→←──2─────────→
 byte byte byte   bytes
```

EOT
SYN: Bit configuration specified by user.

* : This byte is used to store the ENQ when the CHGNTRY macro specifies SKIP. (EOT SYN replaces ENQ, Index.)

Offset count: Used by BTAM to find the beginning of the list.

Note: The symbol specified in the DFTRMLST "Name Field" is used by the assembler as the symbolic address of the first byte of the first entry in the list.

APPENDIX B: HEXADECIMAL EQUIVALENTS FOR POLLING AND ADDRESSING CODES

The following table contains the hexadecimal representation of all alphabetic and numeric characters as they appear in the various transmission codes. These hexadecimal values can be used in the DFTRMLST macro instruction.

Example: To send A over a line, the user must code

62 for a 1030, 1050, 1060, or 2740
18 for a 115A or a 83B3
83 for a Model 33 or 35 TWX Station
A1 for a 2260-2848

Character	1030, 1050, 1060, & 2740	83B3 & 115A*	TWX
A	62	18	83
B	64	13	43
C	67	0E	C3
D	68	12	23
E	6B	10	A3
F	6D	16	63
G	6E	0B	E3
H	70	05	13
I	73	0C	93
J	43	1A	53
K	45	1E	D3
L	46	09	33
M	49	07	B3
N	4A	06	73
O	4C	03	F3
P	4F	0D	0B
Q	51	1D	8B
R	52	0A	4B
S	25	14	CB
T	26	01	2B
U	29	1C	AB
V	2A	0F	6B
W	2C	19	EB
X	2F	17	1B
Y	31	15	9B
Z	32	11	5B

*Keyboards A and C for 83B3 and 115A

Character	1030, 1050, 1060, & 2740	83B3 & 115A*	TWX
1	02	3D	8D
2	04	39	4D
3	07	30	CD
4	08	2A	2D
5	0B	21	AD
6	0D	35	6D
7	0E	3C	ED
8	10	2C	1D
9	13	23	9D
0	15	2D	0D
Space	01	04	05
/	23	-	-

*Keyboards A and C for 83B3 and 115A

Note: The 1050 polling and addressing codes can be sent in either upper or lower case letters but only the lower case is shown.

To poll or address the 2848 Display Control, the hexadecimal values used are:

40 through 5F
A0 through BF
E0 through FF

The valid hexadecimal values for selecting the 2260 Display Station are:

A0 through B8

Macro	Operand	Values Allowed	
ASMTRTAB	tablename	code	RC30,RC50,RF50,RASA,RSCI,RCT1,RCT2,RC80,RC60, RC40,RF40, SD30,SD50,SASA,SSCI,SCT1,SCT2,SD80,SD40,SD60
BTMOD	SEPASMB	code	NO,YES
	BUFFER	code	NO,YES,REQREL
	ERLOGIC	code	E,N,C,NC
	TERMTST	code	NO,YES
	SWITCH	code	NO,YES
	L2260	code	NO,YES
	AUDIO	code	NO,YES
	CANCEL	code	NO,YES
	TRANSL	code	NO,YES
	BSCS	code	NO,YES
	SSAPL	code	NO,YES
	BSCMPT	code	NO,YES
CHGNTRY	list	symbol (r)	Specifying address of the list.
	listype	code	OPENLST,WRAPLST,DIALST,AUTOLST,AUTOWLST, SSALST,SSAWLST
	position	absexp (r)	0-31 for OPENLST,WRAPLST,DIALST 0-97 for AUTOLST,AUTOWLST 0-98 for SSALST,SSAWLST
	numchars	absexp (r)	Specifying the number of polling/ addressing characters per entry.
	action	code	SKIP,ACTIVATE
CHGNTRY (2260 Local)	dtfbt	symbol (r)	Specifying address of the DTFBT.
	listype	code	ATTLST
	position	absexp (r)	0-30 possible, must not exceed number of lines in DTFBT, less one.
	action	code	SKIP,ACTIVATE
CLOSE	dtfname	symbol (r)	1 to 16 addresses may be specified.

Macro	Operand	Values Allowed	
CONTROL*	decb	symbol (r) (1)	Specifying the address of a DECB.
	optype	code	TI,TD,TM
	dtfbt	symbol (r)	Specifying the address of a DTFBT.
	area	relexp (r)	Specifying the address of the I/O area.
	length	absexp (r)	Range allowed: 2-32,767 inclusive.
	entry	relexp (r)	Specifying the address of an entry in a terminal list.
	rln	absexp (r)	The range allowed is 0-30 inclusive.
	MODE	0 1 (r)	Specifying the checking mode to be used.
	MF	code	L,E
DFTRMLST	listype	code	OPENLST,WRAPLST,DIALST,IDLST,AUTOLST,AUTOWLST, SSALST,SSAWLST
	device dependent operands	Note: These operands are defined in the sections describing the individual devices. Some commonly shared values are: • number of dial characters - not to exceed 127. • actual dial characters - can be any decimal digits 0 to 9 inclusive. • number of entries - up to 32 for OPENLST,WRAPLST,DIALST - up to 98 for AUTOLST,AUTOWLST - up to 99 for SSALST,SSAWLST	
DTFBT (Maximum of 255 per assembly)	LINELST	integer	Allowable range: 000-244. (Maximum of 31 may be specified)
	SWITCH	code	NO,YES
	CU	code	2701,2702,2703,7770,7772,2848
	DEVICE	code	1030,1050,1060,2260,83B3,115A,TW33,S360 TW35,2740,2848,2780,1130,2020
	FEATURE	code	PCI,STC,CHK,TRC,BSC,SIX,SXW,SIW,RIX RWX,RIW,MAS,SLV,APL,KBL,OIU
	BUFCB	symbol	Specifying the address of a buffer control block.
	BUFNO	absexp	Allowable range: 2-255 inclusive.
	BUFL	absexp	Allowable range: 16-32,760 inclusive.
	SEPASMB	code	NO,YES

Macro	Operand	Values Allowed	
	MODNAME	symbol	Specifying the name (address) of the BTAM module used. The default name is IJLBTM.
	LERBADR	symbol	Line error block address.
	MODELST	code	0,1,2,3,4,5,6,7
	MSGL	absexp	Allowable range: 2-32,767 inclusive.
	TERMTST	code	NO,YES
	ERROPT	code	E,N,R,W,RW
	CTLCHAR	code	EBCDIC,ASCII,TRNSCD
	RETRY	absexp	Allowable range: 0-15
	CONFIG	code	PPT,MPT
DTFBTND	SEPASMB	code	NO,YES
LOPEN	decb	symbol (r)	1 address may be specified.
OPEN	dtfname	symbol (r)	1 to 16 addresses may be specified.
LERB	nlines	absexp	1 to 31 lines.
	transmct	absexp	1 to 255, default 255.
	datack	absexp	1 to 255, default 10 or 'transmct' if less than 10.
	intreq	absexp	1 to 255, default 5 or 'transmct' if less than 5.
	nontto	absexp	1 to 255, default 5 or 'transmct' if less than 5.
LERPRT	dtfbt	symbol (r) (1)	Specifying the address of the DTFBT.
	rln	absexp (r) (0)	Range allowed: 0 to 30 inclusive. If omitted, all lines having at least one non-zero accumulator will be printed.
	CLEAR	code	NO,YES

Macro	Operand	Values Allowed	
READ*	decb dtfbt length rln MF	Same as for CONTROL.	
	optype	code	TI,TIR,TIL,TT,TTR,TV,TVR,TP,TPR,TB,TBL,TO,TOL,TS,TQ, TIQ,TTL,TPL,TTS
	area	relexp (r) 'S'	Specifying the address of the input area.
	entry	relexp (r) 'S'	Specifying the address of an entry in a terminal list.
RELBUF	dtfbt	symbol (r) (1)	Specifying the address of the DTFBT.
	(r)		Specifying the register (2-12) in which the address of the first buffer in the chain of buffers has been loaded by the user.
REQBUF	dtfbt	symbol (r) (1)	Specifying the address of the DTFBT.
	(r)		Specifying the register (2-12) into which the address of the first buffer in a chain is to be loaded by BTAM.
	count	absexp (r) (0)	Range allowed: 1-255 inclusive.
RESETPL	decb	symbol (r) (1)	Specifying the address of a DECB.
TRNSLATE	dtfbt	symbol (r) (1)	Specifying the address of a DTFBT.
	table**	code (r)	RC30,RC50,RF50,RASA,RSCI,RCT1,RCT2,RC80,RF40, RC40,RC60, SD30,SD50,SASA,SSCI,SCT1,SCT2,SD80,SD40,SD60
	area	relexp (r)	Specifying the address of the I/O area containing the data to be translated.
	length	absexp (r) (0) 'S'	Range allowed: 1-32,767.

Macro	Operand	Values Allowed	
TWAIT	(r)		Specifying the register (2-12) into which BTAM is to load the address of the ECB which was posted complete.
	TERMTST		Coded as shown
	ECBLIST	relexp (r) (1)	Specifying the address of a parameter list.
WAIT	count	absexp (r) (0)	Range allowed: 0-4095 inclusive.
	ECB	symbol (r) (1)	Specifying the address of an ECB.
	ECBLIST	relexp (r) (1)	Specifying the address of a parameter list.
WRITE*	decb dtfbt area rln MF		Same as for CONTROL.
	optype	code	TI,TIR,TIL,TIO,TC,TCO,TT,TTR,TV,TVR,TL,TLR,TLL,TS,TSR, TSL,TB,TA,TN,TIE,TIX,TIXV,TIV,TE,TX,TXV,TD,TR,TW,TQ
	length	absexp (r) 'S'	Range allowed: 2-32,767
	entry	relexp (r) 'S'	Specifying the address of a terminal list or an entry in a terminal list or the address of a parameter list (BSC), or the address of a frame-change control character string (2760 OIU)

*For MF=L: no register notation is permitted; MF operand omitted: register notation not permitted for decb operand.

**A symbol may also be used to specify the address of a user-defined translation table.

APPENDIX D: DOS/BTAM START STOP SAMPLE PROGRAM

```
SMPG   TITLE'SAMPLE PROGRAM USING DOS/BTAM'
***********************************************************************
***                                                                 ***
***    THIS SAMPLE PROGRAM IS DESIGNED TO POLL 2 IBM 1050'S ON A    ***
***    DEDICATED LINE,TO DETERMINE WHICH TERMINAL HAS A MESSAGE      ***
***    TO BE READ,AND TO WRITE THIS MESSAGE TO THE OTHER TERMINAL.  ***
***    THE BTAM MACRO INSTRUCTIONS USED ARE                         ***
***                           BTMOD                                 ***
***                           DTFBT                                 ***
***                           DFTRMLST                              ***
***                           OPEN                                  ***
***                           CLOSE                                 ***
***                           READ                                  ***
***                           WRITE                                 ***
***                           WAIT                                  ***
***                           REQBUF                                ***
***                           RELBUF                                ***
***                                                                 ***
***********************************************************************
       START X'2800'
       USING *,BASERG         ESTABLISH ADDRESSIBILITY
OPEN1  OPEN  DTFBT1           OPEN LINE GROUP FOR USE
REQB1  REQBUF  DTFBT1,(9),5   REQUEST 5 BUFFERS
       BAL   10,TBUFR   CHECK TO SEE IF BUFFERS WERE AVAILABLE
       LA    9,12(9)    POINT REGISTER 9 TO DATA PORTION OF BUFFER
***                                                                 ***
***     POLL BOTH TERMINALS BY USING A WRAP AROUND POLLING LIST     ***
***    READ DATA INTO DATA FIELD OF FIRST BUFFER                    ***
***                                                                 ***
READ1  READ  DECB1,TI,DTFBT1,(9),24,POLL2,MF=E
       BC    7,DUMP1    CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
REQBZ  REQBUF  DTFBT1,(8),6   REQUEST 6 ADDITIONAL BUFFERS
       BAL   10,TBUFR   CHECK TO SEE IF BUFFERS WERE AVAILABLE
       LA    8,12(8)    POINT REGISTER 8 TO DATA PORTION OF BUFFER
       LA    DECBRG,DECB1   LOAD DECB ADDRESS INTO REGISTER
       WAIT  1,ECB=DECB1    WAIT FOR COMPLETION OF READ
       CLI   0(DECBRG),X'7F'  CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL I/O                           ***
       BE    READC       IF GOOD READ,CONTINUE TO READ
READR  READ  DECB1,TP,MF=E   IF ERROR,TRY READING AGAIN
       BC    7,DUMP1    CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
       WAIT  1,ECB=DECB1      WAIT FOR COMPLETION OF READ
       CLI   0(DECBRG),X'7F'  CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL COMPLETION                    ***
       BNE   DUMP1   DUMP IF CONPLETION CODE INDICATES ABNORMAL I/O
***                                                                 ***
***    READ DATA INTO FIRST BUFFER OF SECOND SET                    ***
***                                                                 ***
READC  READ  DECB1,TT,(8),24,MF=E       CONTINUE TO READ
       BC    7,DUMP1   CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
       WAIT  1,ECB=DECB1     WAIT FOR COMPLETION OF READ
       CLI   0(DECBRG),X'7F' CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL I/O                           ***
       BNE   READ2    IF ABNORMAL COMPLETION,TRY READING AGAIN
       B     WRITEA   BRANCH TO WRITE ACK FOR NORMAL COMPLETION
READ2  READ  DECB1,TP,MF=E    IF ERROR,TRY READING AGAIN
       BC    7,DUMP1   CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
       WAIT  1,ECB=DECB1     WAIT FOR COMPLETION OF READ
       CLI   0(DECBRG),X'7F' CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL I/O                           ***
       BNE   DUMP1   DUMP IF COMPLETION CODE INDICATES ABNORMAL I/O
WRITEA WRITE DECB1,TA,MF=E   WRITE POSITIVE ACKNOWLEDGEMENT FOR READ
       BC    7,DUMP1   CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
       LA    DTFREG,ADDRB1  LOAD TERMINAL B1 ADDRESS INTO DTF REGISTER
       WAIT  1,ECB=DECB1     WAIT FOR COMPLETION OF WRITE
       CLI   0(DECBRG),X'7F' CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL I/O                           ***
       BNE   WRITEN   WRITE NAK IF COMPLETION CODE INDICATES
***                            ABNORMAL I/O                         ***
       L     LOCREG,DECB1+20  LOAD ADDRESS OF DEVICE COMMAND CODE
***                            FROM DECB                            ***
       CLI   0(LOCREG),X'FZ'  DID TERMINAL A1 SUPPLY THE INPUT
       BE    WRITE1    IF TRUE,THEN WRITE TO TERMINAL B1
       LA    DTFREG,ADDRA1   OTHERWISE,WRITE TO TERMINAL A1
```

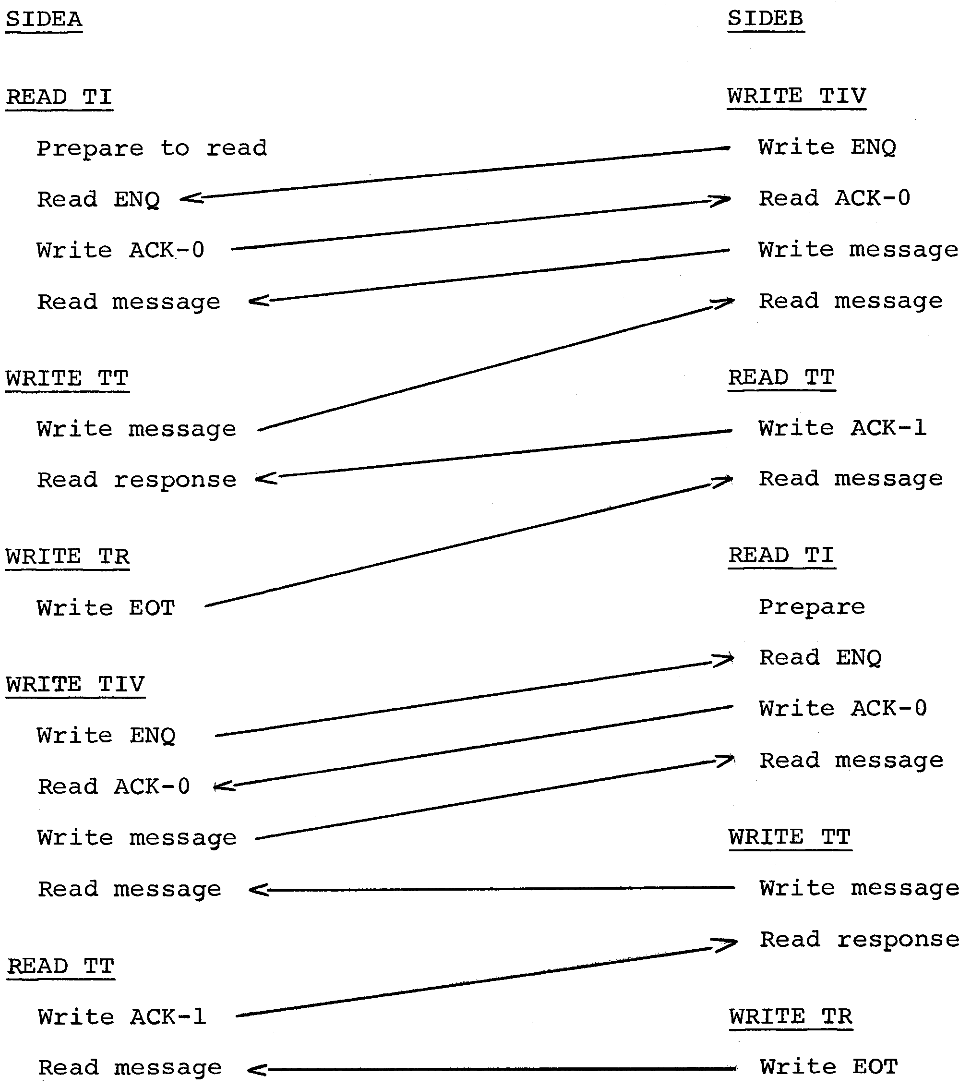
```
WRITE1 WRITE DECB1,TI,DTFBT1,(9),24,(DTFREG),MF=E    WRITE MESSAGE
       BC    7,DUMP1   CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
       WAIT  1,ECB=DECB1     WAIT FOR WRITE COMPLETION
       CLI   0(DECBRG),X'7F' CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL I/O                           ***
       BNE   DUMP1  DUMP IF COMPLETION CODE INDICATES ABNORMAL I/O
WRITEC WRITE DECB1,TT,DTFBT1,(8),24,MF=E    CONTINUE WRITING
       BC    7,DUMP1   CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
       SH    9,=H'12'  POINT REGISTER 9 TO BEGINNING OF BUFFER
       RELBUF DTFBT1,(9)    RELEASE FIRST GROUP OF INPUT BUFFERS
       EAL   10,TBUF   CHECK TO SEE IF BUFFERS WERE RELEASED
       WAIT  1,ECB=DECB1   WAIT FOR WRITE I/O COMPLETION
       CLI   0(DECBRG),X'7F' CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL I/O                           ***
       BE    WRITEN    IF WRITE IS GOOD,GO TO WRITE NAK
       WRITE DECB1,TT,DTFBT1,MF=E   TRY TO WRITE AGAIN
       BC    7,DUMP1    CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
       WAIT  1,ECB=DECB1    WAIT FOR WRITE I/O COMPLETION
       CLI   0(DECBRG),X'7F' CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL I/O                           ***
       BNE   DUMP1   DUMP IF COMPLETION CODE INDICATES ABNORMAL I/O
WRITEN WRITE DECB1,TN,MF=E   WRITE NAK
       BC    7,DUMP1    CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
       SH    8,H'12'   POINT REGISTER 8 TO BEGINNING OF BUFFER AREA
       RELBUF DTFBT1,(8)   RELEASE LAST GROUP OF INPUT BUFFERS
       EAL   10,TBUFR   CHECK TO SEE IF BUFFERS WERE RELEASED
       WAIT  1,ECB=DECB1     WAIT FOR WRITE I/O COMPLETION
       CLI   0(DECBRG),X'7F' CHECK COMPLETION CODE IN DECB FOR
***                            NORMAL I/O                           ***
       BNE   DUMP1  DUMP IF COMPLETION CODE INDICATES ABNORMAL I/O
CLOSE1 CLOSE DTFBT1   CLOSE LINE GROUP FROM FURTHER USE
       EOJ            SPECIFY END OF JOB
TBUFR  LTR   15,15
       BZ    0(10)
DUMP1  PDUMP BEGIN,PRGEND
       B     CLOSE1    GO TO CLOSE DTFBT AND CANCEL JOB
***                                                                 ***
***REGISTER EQUATES***                                              ***
***                                                                 ***
DECBRG FQU   2
DTFREG EQU   3
BASERG EQU   5
LOCREG EQU   6
***                                                                 ***
***DTFBT SYSTEM CONFIGURATION***                                    ***
DTFBT1 DTFBT LINELST=(004),SWITCH=NO,CU=2701,DEVICE=1050,             X
             BUFCB=BFPOOL,BUFNO=12,BUFL=36,SEPASMB=NO,
             MODNAME=IJLBTMAX
***                                                                 ***
***TERMINAL LISTS***                                                ***
***                                                                 ***

POLL2  DFTRMLST  WRAPLST,(E20B,E40B)    POLL A1 AND B1
ADDRA1 DFTRMLST  OPENLST,(E202)
ADDRB1 DFTRMLST  OPENLST,(E204)
***
***READ TO CREATE DECB***
       READ  DECB1,TI,DTFBT1,,,,0,MF=L
       LTORG
ENDSECT CSECT
PRGEND DC    F'0'
       END   BEGIN
/*
```

APPENDIX E: DOS/BTAM BINARY SYNCHRONOUS SAMPLE PROGRAM

```
***                                                                 ***
***********************************************************************
***                                                                 ***
***            THIS BINARY SYNCHRONOUS PROGRAM IS DESIGNED          ***
***            TO ILLUSTRATE COMMUNICATION BETWEEN TWO              ***
***            SYSTEM/360 MACHINES.A 2701 DATA ADAPTER UNIT         ***
***            IS USED IN THIS EXAMPLE AS THE CONTROL UNIT          ***
***            FOR THE TERMINAL CPU.THIS PROGRAM IS DESIGNED        ***
***            SO THAT ANY NUMBER OF MESSAGES CAN BE SENT           ***
***            BETWEEN THE TWO CPU'S.ALL GOOD MESSAGES .            ***
***            RECEIVED BY EACH SYSTEM/360 ARE PRINTED OUT          ***
***            ON A PRINTER                                         ***
***********************************************************************
***                                                                 ***
         START X'2800'
         BALR  5,0
         USING *,5                  ESTABLISH ADDRESSIBILITY
STARTA   OPEN  DTFBTA               OPEN BTAM LINE GROUP FOR USE
         OPEN  MSPRINT              OPEN PRINTER LINE GROUP FOR USE
         SR    CNTRG,CNTRG          CLEAR COUNTER REGISTER
         BAL   RETRG,UPDATE         UPDATE MESSAGE NUMBER
***               READ MESSAGE FROM OTHER TERMINAL                  ***
READTI   READ  DECBA,TI,DTFBTA,INPUTA,LENGTHA,,0,MF=E
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA            WAIT FOR READ COMPLETION
         CLI   DECBA,X'7F'  CHECK COMPLETION CODE IN DECB FOR
***                                NORMAL COMPLETION                ***
         BNE   DUMPA                DUMP IF ABNORMAL COMPLETION
         CLC   INPUTA(4),ENDMSG   CHECK TO SEE IF MESSAGE READ IS
***                                        LAST MESSAGE             ***
         BE    END         IF YES,PRINT MESSAGE ON CONSOLE AND DUMP
         MVC   MSGOUT(47),INPUTA+1   SET UP MESSAGE FOR PRINTER
         PUT MSPRINT,MSGOUT   PRINT INPUT MESSAGE
         CNTRL MSPRINT,SP,2      SPACE PRINTER
         PRTOV MSPRINT,12        CHECK FOR PRINTER OVERFLOW
         WRITE DECBA,TT,,MSG1A,49,MF=E
***        WRITE THE MESSAGE TO THE OTHER TERMINAL AND READ RESPONSE
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA            WAIT FOR WRITE COMPLETION
         CLI   DECBA,X'7F'   CHECK COMPLETION CODE IN DECB FOR
                                   NORMAL COMPLETION
         BNE   DUMPA                DUMP IF ABNORMAL COMPLETION
         WRITE DECBA,TR,MF=E  SEND EOT
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA            WAIT FOR WRITE COMPLETION
         CLI   DECBA,X'7F'  CHECK COMPLETION CODE IN DECB FOR
***                                NORMAL COMPLETION                ***
         BNE   DUMPA                DUMP IF ABNORMAL COMPLETION
         BAL   RETRG,UPDATE         UPDATE MESSAGE NUMBER
         WRITE DECBA,TIV,MSG1A,49,INLISTA,0,MF=E
***      WRITE MESSAGE AND THEN READ MESSAGE FROM OTHER TERMINAL    ***
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA            WAIT FOR WRITE COMPLETION
         CLI   DECBA,X'7F'          CHECK COMPLETION CODE IN DECB FOR
***                                NORMAL COMPLETION                ***
         BNE   DUMPA                DUMP IF ABNORMAL COMPLETION
         MVC   MSGOUT(47),INPUTA+1  SET UP MESSAGE FOR PRINTER
         PUT   MSPRINT,MSGOUT PRINT OUTPUT MESSAGE
         CNTRL MSPRINT,SP,2         SPACE PRINTER
         PRTOV MSPRINT,12           CHECK FOR PRINTER OVERFLOW
         READ  DECBA,TT,INPUTA,55,,0,MF=E    WRITE POSITIVE
***      ACKNOWLEDGEMENT FOR PREVIOUS MESSAGE AND READ NEXT MESSAGE ***
         BC    7,DUMPA  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA            WAIT FOR READ COMPLETION
         CLI   DECBA,X'7F'          CHECK COMPLETION CODE IN DECB FOR
***                                NORMAL COMPLETION                ***
         BNE   DUMPA                DUMP IF ABNORMAL COMPLETION
         BAL   RETRG,UPDATE         UPDATE MESSAGE NUMBER
         B     READTI               CONTINUE READING
END      EQU   *
         READ DECB,TT,,INPUTA,,,,MF=E      READ RESPONSE
         BC     7,DUMPA CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBA        WAIT FOR READ COMPLETION
         EXCP  CCBWRITE EXECUTE CHANNEL PROGRAM TO WRITE OUT MESSAGE
***                          ON CONSOLE
         WAIT  CCBWRITE  WAIT FOR COMPLETION OF CHANNEL PROGRAM
DUMPA    EQU   *
         CLOSE DTFBTA               CLOSE BTAM LINE GROUP
         CLOSE MSPRINT   CLOSE PRINTER LINE GROUP
         PDUMP STARTA,FINISHA DUMP
         EOJ                        SPECIFY END OF JOB
***                                                                 ***
***********************************************************************
***                                                                 ***
***            THE UPDATE ROUTINE IS DESIGNED TO UPDATE THE         ***
***            MESSAGE NUMBER THAT APPEARS IN EACH MESSAGE
***            FORMAT ON THE PRINTER. THIS MESSAGE NUMBER IS
***            THE EXACT NUMBER OF MESSAGES THAT HAVE BEEN
***            READ FROM THE OTHER TERMINAL.
***
***********************************************************************
***                                                                 ***
UPDATE   LA    CNTRG,1(CNTRG)
         CVD   CNTRG,WORKA          CONVERT TO DECIMAL
         UNPK  NUMOUT(4),WORKA(8)   UNPACK
         MVI   NUMOUT,X'40'         MOVE SPACE INTO FIRST POSITION
         OI    NUMOUT+3,X'F0' REMOVE SIGN FROM MESSAGE FORMAT
         BR    RETRG                RETURN TO CALLER
DTFBTA   DTFBT DEVICE=S360,LINELST=(004),CU=2701,MODELST=(2),         X
               CTLCHAR=EBCDIC,FEATURE=(BSC),MODNAME=IJLBTMAX
***            READ TO INITIALIZE DECB
         READ  DECBA,TI,DTFBTA,INPUTA,30,,0,MF=L
***            DOS MACRO TO GENERATE PRINTER MODULE LOGIC
         PRMOD CONTROL=YES,PRINTOV=YES,RECFORM=FIXUNB,WORKA=YES
MSPRINT DTFPR CONTROL=YES,DEVADDR=SYSLST,IOAREA1=MSGPRT,PRINTOV=YES,   -
               WORKA=YES
MSG1A    EQU   *
         DC    X'02'
         DC    C'BINARY SYNCHRONOUS COMMUNICATION -- MESSAGE'
NUMOUT   DC    XL4'00'
         DC    X'03'
INPUTA   DS    80X
LENGTHA  EQU   55
CNTRG    EQU   6
RETRG    EQU   8
ENDMSG   DC    X'02'
         DC    C'END'
WORKA    DS    1D
CCBWRITE CCB   SYSLOG,CCWRITE
CCWRITE  CCW    X'09',MESSAGE,X'20',20
MESSAGE  DC    C'END MESSAGE RECEIVED'
INLISTA  DC    A(INPUTA,55)
MSGOUT   DC    CL132' '
MSGPRT   DC    CL132'  '
FINISHA  DS    1F
         END
         START X'2800'
         BALR  4,0
         USING *,4                 ESTABLISH ADDRESSABILITY
STARTB   EQU   *
         OPEN  DTFBTB              OPEN BTAM LINE GROUP FOR USE
         OPEN  MSPRNTB             OPEN PRINTER LINE GROUP FOR USE
         LA    5,12
         SR    6,6                 CLEAR COUNTER REGISTER
         BAL   RETREG,UPDATEB UPDATE MESSAGE NUMBER
***      WRITE MESSAGE TO AND READ MESSAGE FROM TERMINAL A              ***
WRITIV   WRITE DECBB,TIV,DTFBTB,MSG1B,49,INLISTB,0,MF=E
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB           WAIT FOR COMPLETION OF WRITE
         CLI   DECBB,X'7F'         CHECK COMPLETION CODE IN DECB FOR
***                                 NORMAL COMPLETION                   ***
         BNE   DUMPB               DUMP IF ABNORMAL COMPLETION
         MVC   MSGOUTB(47),INPUTB+1  SET UP MESSAGE FOR PRINTER
         PUT    MSPRNTB,MSGOUTB  PRINT RECEIVED MESSAGE
         CNTRL MSPRNTB,SP,2  SPACE PRINTER
         PRTOV MSPRNTB,12         CHECK FOR PRINTER OVERFLOW
***       WRITE POSITIVE ACKNOWLEDGEMENT AND READ MESSAGE               ***
         READ  DECBB,TT,INPUTB,55,,0,MF=E
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB           WAIT FOR READ COMPLETION
         CLI   DECBB,X'7F'         CHECK COMPLETION CODE IN DECB FOR
***                                 NORMAL COMPLETION                   ***
         BNE   DUMPB               DUMP IF ABNORMAL COMPLETION
         BAL   RETREG,UPDATEB  UPDATE MESSAGE NUMBER
***                                PREPARE AND READ MESSAGE             ***
         READ  DECBB,TI,,INPUTB,55,,0,MF=E
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB           WAIT FOR READ COMPLETION
         CLI   DECBB,X'7F'         CHECK COMPLETION CODE IN DECB FOR
***                                 NORMAL COMPLETION                   ***
         BNE   DUMPB               DUMP IF ABNORMAL COMPLETION
         MVC   MSGOUTB(47),INPUTB+1  SET UP MESSAGE FOR PRINTER
         PUT   MSPRNTB,MSGOUTB  PRINT RECEIVED MESSAGE
         CNTRL MSPRNTB,SP,2     SPACE PRINTER
         PRTOV MSPRNTB,12         CHECK FOR PRINTER OVERFLOW
         WRITE DECBB,TT,,MSG1B,49,MF=E  WRITE MESSAGE TO TERMINAL A
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB           WAIT FOR WRITE COMPLETION
         CLI   DECBB,X'7F'         CHECK COMPLETION CODE IN DECB FOR
***                                 NORMAL COMPLETION
         BNE   DUMPB               DUMP IF ABNORMAL COMPLETION
         WRITE DECBB,TR,MF=E  SEND EOT
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB           WAIT FOR WRITE COMPLETION
         CLI   DECBB,X'7F'         CHECK COMPLETION CODE IN DECB FOR
***                                 NORMAL COMPLETION                   ***
         BNE   DUMPB               DUMP IF ABNORMAL COMPLETION
         BAL   RETREG,UPDATEB UPDATE MESSAGE NUMBER
         BCT   5,WRITIV
         LA    5,12
EXCPWRIT EXCP  CCBWRIT1            EXECUTE CHANNEL PROGRAM TO WRITE OUT
***                                 MESSAGE ON CONSOLE                  ***
         WAIT  CCBWRIT1  WAIT FOR COMPLETION OF CHANNEL PROGRAM
         CLI   MSINPUT,X'C3'  CHECK TO SEE IF THERE ARE MORE
***                                  MESSAGES TO WRITE                  ***
         BE    WRITIV              IF SO,CONTINUE WRITING
         CLI   MSINPUT,X'83'  ARE THERE MORE MESSAGES TO WRITE
         BE    WRITIV              IF SO,CONTINUE WRITING
         CLI   MSINPUT,X'E2'  IS STOP INDICATED
         BE    DUMPB1              IF YES,CONCLUDE CHANNEL PROGRAM AND DUMP
         CLI   MSINPUT,X'A2'  IS STOP INDICATED
         BNE   EXCPWRIT  IF NO,BRANCH TO EXECUTE CHANNEL PROGRAM AGAIN
DUMPB1   EQU   *        DUMP TO CONCLUDE PROGRAM
         WRITE DECBB,TI,,ENDMSG,5,MF=E  WRITE END MESSAGE
         BC    7,DUMPB  CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB           WAIT FOR COMPLETION OF WRITE
         WRITE DECBB,TR,MF=E          SEND EOT
         BC     7,DUMP            CHECK FOR ZERO CONDITION CODE FOR GOOD SIO
         WAIT  ECB=DECBB          WAIT FOR WRITE COMPLETION
DUMPB    EQU   *                   DUMP TO CONCLUDE PROGRAM
         CLOSE DTFBTB              CLOSE BTAM LINE GROUP
         CLOSE MSPRNTB   CLOSE PRINTER LINE GROUP
         PDUMP STARTB,FINISHB  DUMP FROM STARTB TO FINISHB
         EOJ
***                                                                     ***
***********************************************************************
***                                                                     ***
***            THE UPDATE ROUTINE IS DESIGNED TO UPDATE THE             ***
***            MESSAGE NUMBER THAT APPEARS IN EACH MESSAGE
***            FORMAT ON THE PRINTER. THIS MESSAGE NUMBER IS
***            THE EXACT NUMBER OF MESSAGES THAT HAVE BEEN
***            READ FROM THE OTHER TERMINAL.
***
***********************************************************************
***                                                                     ***
UPDATEB  LA    6,1(6)
         CVD   6,WORKB             CONVERT COUNTER TO DECIMAL
         UNPK  OUTB(4),WORKB(8)  UNPACK
         MVI   OUTB,X'40'          MOVE SPACE INTO FIRST POSITION
         OI    OUTB+3,X'F0'        REMOVE SIGN FROM MESSAGE FORMAT
         BR    RETREG              RETURN TO CALLER
DTFBTB   DTFBT DEVICE=S360,LINELST=(005),CU=2701,MODELST=(2)                  X
               CTLCHAR=EBCDIC,FEATURE=(BSC),MODNAME=IJLBTMAX
MSPRNTB  DTFPR CONTROL=YES,DEVADDR=SYSLST,IOAREA1=MSGPRTB,                    X
               PRINTOV=YES,WORKA=YES
***            READ TO INITIALIZE DECB
         READ  DECBB,TI,DTFBTB,INPUTB,30,,0,MF=L
***            DOS MACRO TO GENERATE PRINTER MODULE LOGIC
         PRMOD CONTROL=YES,PRINTOV=YES,RECFORM=FIXUNB,WORKA=YES
MSPRNTB  DTFPR CONTROL=YES,DEVADDR=SYSLST,IOAREA1=MSGPRTB,PRINTOV=YES, -
               WORKA=YES
MSG1B    EQU   *
         DC    X'02'
         DC    C'BINARY SYNCHRONOUS COMMUNICATION -- MESSAGE'
OUTB     DC    XL4'00'
         DC    X'03'
ENDMSG   DC    X'02'
         DC    C'END'
         DC    X'03'
RETREG   EQU   10
MSINPUT  DS    CL4'00'
         SPACE 2
INPUTB   DS    80X
INLISTB  DC    A(INPUTB,55)
WORKB    DS    1D
CCBWRIT1 CCB   SYSLOG,CCWRITE
CCWRITE  CCW   X'09',MESSU,X'60',26
         CCW   X'0A',MSINPUT,0,2
MESSU    DC    C'REPLY C TO WRITE,S TO STOP'
MSGOUTB  DC    CL132'  '
MSGPRTB  DC    CL132'  '
ENDSECT  CSECT
FINISHB  DS    1F
         END
```

The following illustration shows the relationship between the two CPU's of the above binary synchronous program. SIDEA and SIDEB represent the two CPU's.



APPENDIX F: CODE CHARTS FOR BINARY SYNCHRONOUS COMMUNICATION

This appendix includes code charts for EBCDIC, USASCII, and 6-bit TRANSCODE.

SIX BIT TRANSCODE

Code Positions	0 1	0 1	0 1	0 1
2 3 4 5	0 0	0 1	1 0	1 1
0 0 0 0	SOH 12-9-1	& 12	- 11	0 0
0 0 0 1	A 12-1	J 11-1	/ 0-1	1 1
0 0 1 0	B 12-2	K 11-2	S 0-2	2 2
0 0 1 1	C 12-3	L 11-3	T 0-3	3 3
0 1 0 0	D 12-4	M 11-4	U 0-4	4 4
0 1 0 1	E 12-5	N 11-5	V 0-5	5 5
0 1 1 0	F 12-6	O 11-6	W 0-6	6 6
0 1 1 1	G 12-7	P 11-7	X 0-7	7 7
1 0 0 0	H 12-8	Q 11-8	Y 0-8	8 8
1 0 0 1	I 12-9	R 11-9	Z 0-9	9 9
1 0 1 0	STX 12-9-2	SPACE No Punch	ESC 0-9-7	SYN 9-2
1 0 1 1	. 12-8-3	$ 11-8-3	, 0-8-3	# 8-3
1 1 0 0	¤ 12-8-4	* 11-8-4	% 0-8-4	@ 8-4
1 1 0 1	BEL 0-9-8-7	US 11-9-8-7	ENQ 0-9-8-5	NAK 9-8-5
1 1 1 0	SUB 9-8-7	EOT 9-7	ETX 12-9-3	EM 11-9-8-1
1 1 1 1	ETB 0-9-6	DLE 12-11-9-8-1	HT 12-9-5	DEL 12-9-7

STANDARD REPRESENTATION OF USASCII

Rows	Columns	0	1	2	3	4	5	6	7
	$b_7 b_6 b_5$	000	001	010	011	100	101	110	111
	$b_4 b_3 b_2 b_1$								
0	0 0 0 0	NUL	DLE	SP	0	@	P	`	p
1	0 0 0 1	SOH	DC1	!	1	A	Q	a	q
2	0 0 1 0	STX	DC2	"	2	B	R	b	r
3	0 0 1 1	ETX	DC3	#	3	C	S	c	s
4	0 1 0 0	EOT	DC4	$	4	D	T	d	t
5	0 1 0 1	ENQ	NAK	%	5	E	U	e	u
6	0 1 1 0	ACK	SYN	&	6	F	V	f	v
7	0 1 1 1	BEL	ETB	'	7	G	W	g	w
8	1 0 0 0	BS	CAN	(	8	H	X	h	x
9	1 0 0 1	HT	EM	)	9	I	Y	i	y
10	1 0 1 0	LF	SUB	*	:	J	Z	j	z
11	1 0 1 1	VT	ESC	+	;	K	[	k	{
12	1 1 0 0	FF	FS	,	<	L	\	l	¦
13	1 1 0 1	CR	GS	-	=	M	]	m	}
14	1 1 1 0	SO	RS	.	>	N	∧	n	~
15	1 1 1 1	SI	US	/	?	O	_	o	DEL

DATA LINK CONTROL FUNCTIONS

	Characters Used In:		
Function	USASCII	EBCDIC	Transcode
ACK-0	DLE,0	DLE,X'70'	DLE, - (hyphen)
ACK-1	DLE,1	DLE,X'61'	DLE,T
WABT	DLE,?	DLE,X'7F'	DLE,W
SAK	DLE,; (semicolon)	DLE,X'6B'	DLE,Z

Bit Positions 0 and 1 →	00				01				
Bit Positions 2 and 3 →	00	01	10	11	00	01	10	11	Digit Punches
0000	① NUL	② DLE	③ DS	④	⑤ SP	⑥ &	⑦ -	⑧	
0001	SOH	DC1	SOS				⑬ /		1
0010	STX	DC2	FS	SYN					2
0011	ETX	DC3							3
0100	PF	RES	BYP	PN					4
0101	HT	NL	LF	RS					5
0110	LC	BS	EOB	UC					6
0111	DEL	IL	PRE	EOT					7
1000		CAN							8
Zone Punches	9 12	9 11	9 0	9	9 12 0	9 12 11	9 11 0	9 12 11 0	

Bit Positions 4, 5, 6, and 7

Bit Positions 0 and 1 →	10				11				
Bit Positions 2 and 3 →	00	01	10	11	00	01	10	11	Digit Punches
0000					⑨	⑩	⑪	⑫ 0	8-1
0001	a	j			A	J	⑭	1	1
0010	b	k	s		B	K	S	2	2
0011	c	l	t		C	L	T	3	3
0100	d	m	u		D	M	U	4	4
0101	e	n	v		E	N	V	5	5
0110	f	o	w		F	O	W	6	6
0111	g	p	x		G	P	X	7	7
1000	h	q	y		H	Q	Y	8	8
1001	i	r	z		I	R	Z	9	9
Zone Punches	12 0	12 11	11 0	12 11 0	12	11	0		

Bit Positions 4,5,6 and 7

Bit Positions 0 and 1 →	00				01				
Bit Positions 2 and 3 →	00	01	10	11	00	01	10	11	Digit Punches
1001		EM							8-1
1010	SMM	CC	SM		¢	!	⑮	:	8-2
1011	VT	CU1	CU2	CU3	.	$	,	#	8-3
1100	FF	IFS		DC4	<	*	%	@	8-4
1101	CR	IGS	ENQ	NAK	(	)	_	'	8-5
1110	SO	IRS	ACK		+	;	>	=	8-6
1111	SI	IUS	BEL	SUB	|	¬	?	"	8-7
Zone Punches	9 12	9 11	9 0	9	12	11	0		

Bit Positions 4, 5, 6, and 7

Bit Positions 0 and 1 →	10				11				
Bit Positions 2 and 3 →	00	01	10	11	00	01	10	11	Digit Punches
1010									8-2
1011									8-3
1100									8-4
1101									8-5
1110									8-6
1111									8-7
Zone Punches	12 0	12 11	11 0	12 11 0	9 12 0	9 12 11	9 11 0	9 12 11 0	

Bit Positions 4, 5, 6, and 7

①	12-0-9-8-1	④	12-11-0-9-8-1	⑦	11	⑩	11-0	⑬	0-1
②	12-11-9-8-1	⑤	No Punches	⑧	12-11-0	⑪	0-8-2	⑭	11-0-9-1
③	11-0-9-8-1	⑥	12	⑨	12-0	⑫	0	⑮	12-11

USASCII	EBCDIC Equivalents
RS	IRS
ETB	EOB
ESC	PRE
FS	IFS

GLOSSARY OF THE TERMS USED IN DOS/BTAM

Access Method: Any data management facility available to the user for transferring data between main storage and an input/output device.

ACK: Positive Acknowledge character.

Addressing: A technique by which the central processing unit (CPU) prepares a remote station to receive a message.

Addressing Characters: Characters used to establish contact with a terminal.

Answering: A remote station has the ability to call the central processing unit through the switched network facilities. The CPU will answer the call.

Audio Response Unit (ARU): A control unit, such as IBM 7770 or IBM 7772, that is able to deliver an audio response to a digital inquiry.

Basic Access Method: Any access method in which each input/output statement causes a corresponding machine input/output operation to occur.

Baud: A unit of modulation rate for signaling. The speeds in bauds is equal to the number of bits per second transmitted, including start bits and stop bits, if present (see Start-Stop Transmission).

Baudot Code: A five-level transmission code.

Basic Telecommunications Access Method (BTAM): A basic access method for communication with terminals.

Block: See Message Block.

Buffer (program input/output): A portion of main storage into which data is read or from which it is written, concurrent with asynchronous CPU activity.

Buffer pool: Main storage area reserved per line group for buffering.

Calling: A switch exchange service which enables a telephone user to select another subscriber, i.e., dialing.

Command Control Block (CCB): Refer to DOS/360 System Control and System Service Programs (Version II).

Chaining: An organization of records in which each record belongs to a group of records and has a linking field for tracing the chain.

Communication Lines: A communication line is the medium (wire, carrier channel, radio channel, etc) over which signals may be sent.

Contention: A condition on a (start-stop) multipoint communication channel when two or more locations try to transmit at the same time (see Polling). In binary synchronous communication, this term denotes the condition on a point-to-point communication line when the CPU and the remote device bid for control of the line simultaneously.

Control Character: A special character sent over a line which is recognized by the control units (control units of the terminals as well as telecommunications control units). Example: End of transmission (EOT), polling characters, addressing characters, etc.

Control Mode: A line is in control mode when all terminals on the line interpret the next characters sent as control characters.

CRT: Cathode Ray Tube.

Cursor: Positional index which indicates the position where the next character will be displayed on the screen of a 2260 Display Unit.

Data Adapter Unit (2701): A control unit that houses transmission adapters. Each transmission adapter contains the circuitry and the logic for the control and logical connection of a terminal. One transmission adapter is required per line. (See Telecommunication Control Unit.)

Data Collection System: A system mainly used to gather data.

Data Communication System: A system mainly used for the transmission of information.

Data Link: The equipment that permits transfer of data between two stations.

Data Set: A modulator/demodulator device designed to provide compatibility between input/output equipment and communication facilities (see Modem).

Dedicated System: When the main storage requirements of the Teleprocessing program does not permit another job to run simultaneously, the system is dedicated to the Teleprocessing program and the Teleprocessing program is run as a background program.

Digitally Coded Voice (DCV): Binary information necessary to the IBM 7772 Audio Response Unit to synthetize back the sound, word, or sentence so coded, when received from the CPU through a WRITE command. The digital equivalent of all the sounds, words, or sentences used by a given application will usually be stored on a direct access device. An average of 300 bytes per second of audio output is necessary.

End of Block (EOB): A control character used to partition a message into blocks and initiate a control function.

End of Inquiry: Any character recognized by the IBM 7770 and IBM 7772 as a signal that the inquiry is complete and the end of read command status should be generated and sent to the CPU. The character itself will be sent as the last character of the input message.

End of Transmission (EOT): A control character (or character string) that specifies the end of a message and causes a reset of all stations.

Event (in BTAM): An event is the satisfying of a WAIT condition, and may be one of the following:

1. a successful completion of a READ, WRITE, CONTROL, or RESETPL operation;

2. a PCI interruption (7772 only); or

3. an unsuccessful operation, as follows:

    a. polling--when the end of an OPEN list is reached, polling stops;

    b. addressing--when the addressed device is not ready, no message is sent;

    c. multi-addressing--when one of the addressed devices is not ready, the message is not sent to the ready devices; and

    d. an error condition.

Inquiry: An input message calling for an answer.

Leased Line: A data path reserved for the exclusive use of a customer.

Line: See Communication Line.

Line Connection: For terminals attached to a switched network, a physical connection has to be established before data transmission can take place.

Line Control Characters: A set of special characters which are not part of the text. A line control character is recognized by a telecommunications control unit, which performs the function associated with that character.

Local Device: A local device is a device directly attached to a channel by the means of a control unit (see Remote Device).

LRC Character: Abbreviation for Longitudinal Redundancy Checking character.

Message: Text terminated by an End-of-Transmission Character (EOT).

Message Block: A portion of a message terminated by an End-of-Block (EOB) character (1030, 1050, 1060) or by an End-of-Text (ETX) character (2260).

Message Record: A whole message or a portion of a whole message, received, or sent by a single read or write command. A message record will be terminated by an EOB character (1030,1050,1060), by an ETX character (2260), or by an EOT character (83B3 and 115A).

Message Segment: A portion of a message that is contained in a single buffer.

Modem: Acronym for modulator/demodulator device. (See Data Set.)

Modulation: The periodic variation of amplitude, frequency, or phase of a carrier wave created by the data flow.

Multi-Component Addressing: When the CPU selects several receiving devices of the same terminal and sends a message that is recorded by all selected devices.

Multidrop Line: Same as Multipoint Line.

Multipoint Line: A line with one or more terminals attached to it. (See Point-to-Point.)

Multiterminal Addressing: When the CPU selects several receiving devices pertaining to several terminals on one line and sends a message, the message is recorded by all selected devices.

NAK: Negative Acknowledge character.

Nonswitched Line: A multidrop or point-to-point configuration when line connection is not established through a switched network.

Network: A series of points interconnected by communication channels (see Switched Network).

Point-to-Point: A line configuration is point-to-point when that line permits exchange of information between two stations only. Example: On a switched network, once connection is established, the configuration is point-to-point (see Multidrop).

Polling: The process of inviting stations within a data link to transmit messages. The stations are invited one at a time in an orderly fashion. The basic function of polling is to prevent contention by insuring that only one station transmits at a time. (See Contention.)

Polling Characters: Characters used to establish contact with a terminal.

Private Line: A data path reserved exclusively for one user.

Remote Device: One which is attached to a telecommunication control unit by the means of a communication network.

Start-Stop Transmission: A start bit and a stop bit indicate the beginning and the end of the character.

Station: An aggregate of equipment and control attached to any one of the several ends of a communication line (see Terminal).

Switched Lines: Lines whose connection must be established prior to the start of data transmission.

Switched Network: The connection is established between the calling party and the called party prior to the start of data transmission and is broken at the end of the data transmission.

Telecommunications: A general term expressing data transmission between remote locations.

Teleprocessing: A term associated with IBM telecommunication systems expressing data transmission between a computer and remote devices.

Telecommunications Control Unit: A device used to transmit or adapt messages, coming from a remote device, for the CPU. Included are the 2701, 2702, 2703, 7770, and 7772.

Terminal: Equipment on a communication channel that may be used for either input or output (see Station).

Terminal Component: An input or output device that is part of a terminal (see Station).

Text Mode: A line in text mode, when all receiving units logically connected to the transmitting unit record the next characters sent until an End-of-Transmission EOT character sets the line in control mode.

Transmission Control Unit (2702, 2703): A control unit that houses terminal controls. Each terminal control contains the circuitry and the logic for the control and logical connection of all terminals of the same type within the system (see Data Adapter Unit).

Unit: In Teleprocessing, refers to a physical I/O unit, characterized by a unique physical address. For example, a line is a unit; the IBM 2260 local is a unit.

VCT: A Voice Code Translator is the hardware part of the IBM 7772 (also called output channel) which converts DCV data into audio form.

VRC Character: Abbreviation for Vertical Redundancy checking character.

READER'S COMMENT FORM

IBM System/360
Disk Operating System
Basic Telecommunications Access Method C30-5001-5

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. All comments will be handled on a non-confidential basis. Copies of this and other IBM publications can be obtained through IBM Branch Offices.

	Yes	No
● Does this publication meet your needs?	□	□
● Did you find the material:		
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
Well illustrated?	□	□
Written for your technical level?	□	□

- What is your occupation? ____________________
- How do you use this publication?

As an introduction to the subject?	□	As an instructor in a class?	□
For advanced knowledge of the subject?	□	As a student in a class?	□
For information about operating procedures?	□	As a reference manual?	□

Other ____________________

- Please give specific page and line references with your comments when appropriate.

COMMENTS:

- Thank you for your cooperation. No postage necessary if mailed in the U. S. A.

Staple

YOUR COMMENTS, PLEASE . . .

This publication is one of a series that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in using your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold Fold

FIRST CLASS
PERMIT NO. 569
RESEARCH TRIANGLE PARK
NORTH CAROLINA

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 12275
Research Triangle Park
North Carolina 27709

Attention: Programming Documentation, Dept. 844

Cut Along Line

Fold Fold

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]

Additional Comments:

C30-5001-5

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]