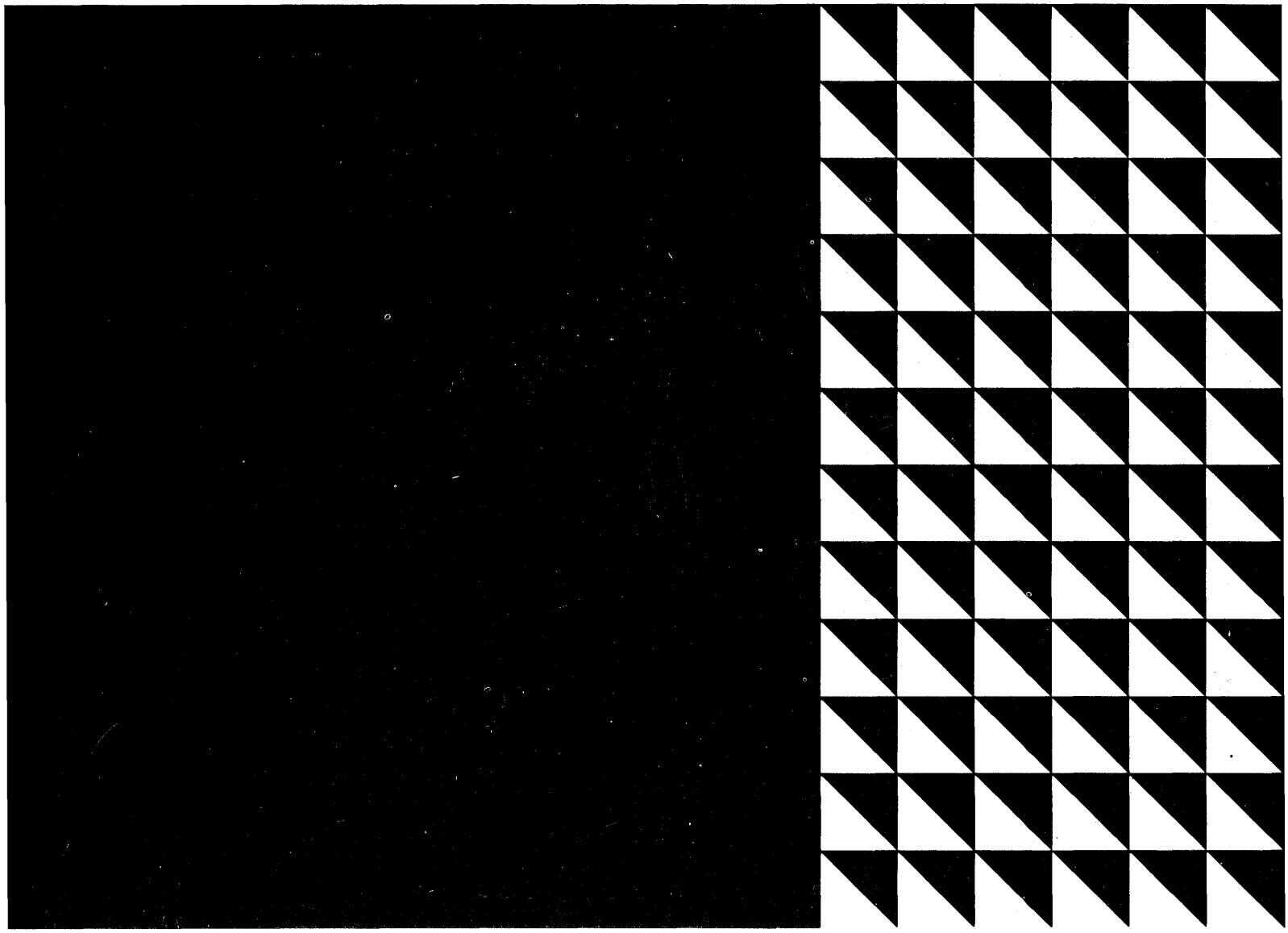
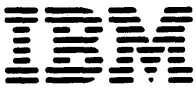
IBM

# IBM 3704 and 3705 Communications Controllers Emulation Programming



Student Text



# IBM 3704 and 3705 Communications Controllers Emulation Programming

Student Text

Minor Revision, September 1978
Incorporates Table of Contents, SN23-1048.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. Address comments concerning the contents of this publication to IBM Corporation, DPD Education Development - Publishing/Media Support, Education Center, South Road, Poughkeepsie, New York 12602.

# Preface

This publication is a student text on the IBM 3704 and 3705 Communications Controllers emulation programming. A knowledge of the hardware is required for coding many of the macro operands which define the network. *IBM 3704 and 3705 Communications Controllers Hardware* (SR20-4544), provides the hardware prerequisite to this material.

A quiz is presented at the end of each major section of this manual. The answers to the quiz are given in Appendix A of the text.

If you need additional information, please refer to:

*Introduction to the IBM 3704 and 3705 Communications Controllers* (GA27-3051)

*IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008) (for VS systems)

*IBM 3704 and 3705 Communications Controllers Emulation Program Generation and Utilities Guide and Reference Manual* (GC30-3002) (for nonVS systems)

*Teleprocessing Preinstallation Guide for IBM 3704 and 3705 Communications Controllers* (GC30-3020)

# Contents

# Emulation Programming

## Objective

Upon completion of this topic the student should be able to identify the terms and concepts of emulation programming, the sequences of macros, the generation process, loading, and dumping of the emulation program.

## Introduction

Note:

The following text refers to prerequisite material on the IBM 3704 and 3705 Communications Controller hardware. This information is available in *IBM 3704 and 3705 Communications Controller Hardware* (SR20-4544)

The coding exercises require the following SRL's:

*IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual (VS)* (GC30-3008)

*IBM 3704 and 3705 Communications Controllers Emulation Program Generation and Utilities Guide and Reference Manual* (GC30-3002), (nonVS)

*Teleprocessing Preinstallation Guide for IBM 3704 and 3705 Communications Controllers* (GC30-3020)

The IBM 3704 and 3705 Communications Controllers are programmed transmission control units that communicate with a variety of terminals, transmission control units, and computers. The programming can accommodate many different telecommunication applications and operational requirements.

The emulation program resides in the IBM 3704 and 3705 Communications Controllers and allows either controller to perform most functions of an IBM 2701 Data Adapter Unit, IBM 2702 Transmission Control unit, IBM 2703 Transmission Control Unit, or any combination of the three. The emulation program allows many programs written for support of the 2701, 2702, and 2703 to operate without modification with either communications controller. Such programs include IBM Type I access methods that support the 2701, 2702, and 2703, as well as IBM Type II and Type III programs and user-written programs that interface in a manner equivalent to IBM Type I access method programs. Programs that involve timing dependencies and support of certain special and custom features may have to be changed.

The emulation program requires that a type 1 or type 4 channel adapter be installed in the controller for attachment to a System/360 or System/370 byte-multiplexor channel. All models of the 3704 and the 3705 have enough storage to accommodate small emulation program configurations, but larger configurations require more than the minimum amount of storage (16K).

The emulation program in conjunction with the type 1 or type 4 channel adapter not only permits the use of the same control sequences and data transfers as the 2701, 2702, and 2703, but also provides most of the standard functions of these control units. Not supported are the parallel data adapter, synchronous data adapter type 1, the programmable two-processor switch, two-channel attachment (except by two type 4 channel adapters), six-bit transcode, 230,400 bps synchronous speed, selector channel attachment, direct attachment of the IBM 1032 Digital Time Unit, and the IBM 2712 Remote Multiplexor attachment features.

In addition to the standard 2701, 2702, and 2703 functions, the emulation program also supports certain RPQ's (request for price quotation) for these control units.

In emulation mode, the 3705 with one channel adapter can attach up to 255 communication lines for half-duplex operation at speeds from 45.5 bps to 56,000 bps; with two type 4 channel adapters up to 352 communication lines are supported. The 3704 can attach up to 32 communication lines at speeds from 45.5 bps to 50,000 bps. The actual number of lines that can be attached depends upon the specific configuration of the telecommunications subsystem.

Each line attached to the controller in emulation mode requires a nonshared subchannel address on the byte-multiplexor channel. A line may be switched between subchannel addresses using the multiple subchannel line access (MSLA) facility. MSLA defines multiple subchannel definitions to one line address. The first subchannel to enable the line gains control of the line until reset by a disable.

Appendix B contains a complete list of the types of stations supported by the communications controller when executing an emulation program.

The emulation program is generated according to the network configuration of each installation. To begin the generation procedure, the user codes a series of macro instruction source statements that define the telecommunication network. The source statements are then generated by a compilation process into an emulation program load module, which is loaded into the communications controller. By dumping the communications controller storage, the user may examine the contents of controller storage as an aid to error diagnostics.

This publication explains how to define, generate, load and dump an emulation program.

## Defining the Emulation Program

Achieving an operating emulation program is a three-step process. The first step, defining the control program, is the most involved. Many different variables and options must be considered in preparing a control program to meet the requirements of a particular network configuration and application.

An emulation program is defined in the form of a source program consisting entirely of macro instructions. These macro instructions include:

1. System macros which provide information pertaining to the entire controller -- hardware features, certain control program options, and program generation information such as data set names.
2. Configuration macros which provide the information necessary to construct tables needed by the emulation program. Such tables are used to control the flow of data between the controller and the telecommunication network and between the host processor and the communications controller.
3. A generation delimiter macro that ends the control program generation input stream.

When punched into cards and preceded by the appropriate job control statements, the source program forms the input to the next step: the generation procedure.

## Generating the Emulation Program

After definition in the form of a source program containing emulation program macro instructions, the emulation program is ready to be generated. The primary output of the generation procedure is an emulation program load module ready for loading into the communications controller.

**Generating the Emulation Program Under VS1, VS2, and MVS** The emulation program generation procedure under the control of VS1, VS2, and MVS is a two-stage process.

In the first stage of the generation procedure, the emulation program generation macros which have been coded are assembled by the communications controller assembler or a host operating system assembler. The output from the assembler is a job stream containing the data and control statements necessary to create the desired emulation program. The job stream is a sequential data set that can be directed to cards, tape, or a direct access storage unit.

Operator intervention is required between the stages of program generation. Diagnostic messages produced at the end of stage 1 indicate any errors that have occurred. If the errors are serious, no job stream is produced; the source statements must be corrected and stage 1 reexecuted. If the errors are not serious, the operator initiates the second stage, specifying as input the stage 1 output.

Using the controller assembler, the second stage of the generation procedure assembles the control tables that are required by the emulation program being generated. Then the linkage editor is executed to combine the appropriate modules into an emulation program load module, which is loaded from the host processor into the communications controller.

**Generating the Emulation Program Under DOS/VS** The emulation program generation procedure under the control of DOS/VS is a two-stage process. In the first stage of the generation procedure, the macros which were coded are assembled by the communications controller assembler. The output from the assembler is an object deck and a printed listing.

Operator intervention is required between the stages of program generation. Diagnostic messages are printed in the stage 1 output listing, indicating any errors that have occurred. If there are no serious errors, an object deck is produced.

Stage 2 of the generation procedure is made up of two substeps. First, the operator catalogs the object deck in a library together with IBM-supplied emulation modules. Second, there is a linkage editing of the modules

according to the INCLUDE cards on the stage 1 output listing. The result is an emulation program load module that is located in the core image library.

### Communications Controller Utilities

The loader, dump, and dynamic dump programs are utilities used with the communications controller. Each utility is initiated by the appropriate job control statements and control cards. Each utility is described briefly below.

**The Loader Utility** The final step in achieving an operating emulation program consists of loading the control program load module from the host processor into the communications controller. This step is accomplished through the use of a loader utility program that is executed in the host processor, with the controller online to the processor.

The loader utility is supplied as one of the emulation program system support programs.

**The Dump Utility** The dump utility program allows all or part of the contents of controller storage to be transferred from the controller to the host processor, which then prints the contents in hexadecimal format. You can request a formatted dump or an unformatted dump of controller storage. The unformatted storage dump prints with EBCDIC equivalents on the right side of the page. The formatted storage dump prints the mnemonic operation codes with the instructions.

Executing the dump utility stops operation of the emulation program. After the dumping process is completed, the emulation program must be reloaded into the controller before telecommunication operations can resume.

**Dynamic Dump Utility** The dynamic dump is an optional utility program that allows the contents of controller storage to be transferred from the controller to the host processor without stopping the operation of the emulation program. A full storage dump or a dump of the trace table can be obtained. In addition, portions of storage can be displayed on the operator's console at the host processor. The utility also can activate or deactivate the emulation program line-trace function and allows the selection of two program levels to be traced.

### Introduction Review

Macros define the emulation program for each network configuration. The macros are assembled in a generation process to create the control blocks for the configuration. The linkage editor combines the control blocks with preassembled modules to produce the emulation program load module.

A loader utility is initiated by job control and utility control statements to transfer the load module to the communications controller. The loader utility is one of the IBM-supplied emulation program system support programs.

A dump of the emulation program may be obtained by either of two methods. A dump utility, initiated by job control and utility control cards, transfers the contents of storage to the host processor for listing. If the dump utility is used, the emulation program is ended and must be reloaded. The dynamic dump utility does not end normal emulation program execution, but allows concurrent emulation and dumping. The dynamic dump can dump storage, trace entries, or trace two program levels.

### Introduction Quiz

The following quiz is provided to enable you to check yourself on this subject matter. The answers are given in Appendix A.

True or False:

1. The emulation program is generated by assembling macros which define the configuration.
2. The dump utility allows the contents of storage to be dumped while the emulation program continues to execute.
3. The loader utility is initiated by job control in the host.
4. The emulation program can operate over a type 1, 2, 3, or 4 channel adapter of a communications controller.
5. The emulation program requires a host connection to a host byte-multiplexor channel.
6. The emulation program emulates the IBM 2701 Data Adapter Unit, IBM 2702 Transmission Control Unit, IBM 2703 Transmission Control Unit, or any combination of the three.

Completion:

7. A communications controller with one type 1 or one type 4 channel adapter and the emulation program can support a maximum of _____ lines.

8. Two host systems can share emulation lines (not concurrently) if the communications controller is equipped with two channel adapters. Both channel adapters must be type _____?

Criterion

If you missed more than two questions, you should review this section.

# Defining the Emulation Program

## Objective

Upon completion of this topic the student should be able to identify the stage 1 macros required for emulation generation, state the macro sequence, and use system reference manuals to code for a defined configuration.

## Introduction

The emulation program must be defined for each specific 3704 or 3705 communications controller configuration. The definition is coded using IBM-supplied generation language macros. The macros are then assembled to create the emulation load module.

**The Control Program Generation Language** The control program generation language provides a high-level method of generating the control program for the 3704 or the 3705. The language is designed to minimize the programming effort for even the most complex configuration of lines and stations.

The emulation generation language is made up of macro instructions that fall into three categories according to the type of parameters to be defined. The types of macros are: (1) system macros, (2) configuration macros, and (3) a generation delimiter macro.

*System Macros*

The system macros provide information pertaining to controller hardware features, certain control program options, and program generation information such as data set names. The system macros specify, for example:

- -- The channel addresses
- -- Type of generation (EP, NCP, PEP)
- -- Type of host operating system (OS, DOS)
- -- Optional trace and test facilities

*Configuration Macros*

The configuration macros provide information for the tables needed by the control program to control the flow of data between the host subchannel and the scanner line interface. These macros define the characteristics of the elements of the telecommunication network -- line groups and communication lines. The macros must be arranged in a specific order to associate a particular communication line with a particular line group. All line macros to be associated with a specific group macro definition must follow that group macro without other macro definitions intervening. For example, two group definitions with three lines in each group are arranged as follows:

Group 1 definition
  Line definition
  Line definition
  Line definition

Group 2 definition
  Line definition
  Line definition
  Line definition

Each macro is associated with the last higher-level macro that precedes it. This type of structure simplifies coding by allowing characteristics that are the same for all lines of a group to be coded on the highest level: the group definition.

There is no requirement for groups to be arranged in any particular sequence.

*Generation Delimiter Macro*

If the 3704 and 3705 assembler is used for stage 1 assembly, the generation delimiter ends the program input stream. The 3704 and 3705 assembler recognizes the GENEND macro as a replacement for the END statement of a control section. If a host assembler is used, an END statement must be added following the generation delimiter.

**Coding the Generation Language** The generation language is designed to make coding as easy as possible. All the operands of the individual macros are keywords, so the programmer does not have to be concerned with the sequence in which the operands are coded. However, the relative order of the macros in the input stream is fixed to some extent.

There are five macros for emulation program generation. The BUILD macro is the first macro of the definition. Immediately following the BUILD macro are one to four CSB macros, one per installed 3704 or 3705 communications scanner. After the last CSB macro are the definitions of lines and line groups. Each line group is defined by a GROUP macro, which is followed immediately by one or more LINE macro definitions. After all line groups and lines are defined, the GENEND macro is coded as the delimiter of the generation definition. Figure 2.1 summarizes the emulation generation macros.

| | |
|---|---|
| BUILD | – The first macro of a generation. |
| CSB | – Defines a communication scanner; one CSB macro per scanner (one to four); immediately follow the BUILD macro. |
| GROUP | – One to 255 line groups; groups lines by general type. |
| LINE | – One to 32 (3704). one to 352 (3705) lines one or more LINE macros follow each GROUP macro. |
| GENEND | – The last macro of a generation. |

**Figure 2.1 The emulation generation macro sequence.**

Coding operands on the macros requires a knowledge of the telecommunication subsystem that is to be defined. The characteristics of the telecommunication subsystem can be divided into the following areas:

1. Stations and lines of the telecommunication network
2. Communications controller hardware configuration
3. Type of transmission control unit to be emulated
4. Procedural options governing message traffic between the controller and the network
5. Optional service aid facilities
6. Program generation options
7. Data sets used during generation

Once you are familiar with the characteristics that apply to your equipment configuration and applications, you are ready to code the program generation macro instructions. The following material provides information on characteristics which you will be required to define in the macro operands.

*Station Characteristics*

The term 'station' refers to any equipment, regardless of type, that can transmit data onto or receive data from a communication line connected to the communications controller. For operations in emulation mode, this definition includes (1) computers, (2) transmission control units such as the IBM 2701, 2702, and 2703, and (3) the input/output units (keyboards, printers, tape and card readers, punches, and display screens) usually referred to as terminals.

*Station Features*

The presence of the features explained below is specified in the FEATURE operand of the LINE macro representing the line over which the controller communicates with the terminal.

Record Checking:

Some start-stop terminals have the record-checking capability (also called longitudinal redundancy checking); others do not. For each line in the network you must specify to the emulation program whether the terminals with which the program communicates over that line have the record-checking capability. If the terminal is an IBM 1050, 1060, 2260, 2845, 2848, or System/7, all of which do have this capability, specify LRC in the FEATURE operand of the LINE macro. Also specify LRC for an IBM 2740 (Model 1 or 2) equipped with the record checking feature.

Downshifting on Space Characters:

Some AT & T 83B3, Western Union 115A, and World Trade teletypewriter terminals, upon sending or receiving a space character, automatically downshift so that subsequent message text is in lower case (downshift) mode. Automatic downshifting avoids the need to send a LTRS character to effect downshifting. In the FEATURE operand of the LINE macro, specify SPACE if the terminals are so equipped.

Immediate End:

Upon receiving an end-of-transmission character from a start-stop terminal, the emulation program normally delays ending the receive operation for several character times (the time required for the transmission of one character) until the line becomes electrically 'quiet'. The absence of further characters following the EOT verifies the EOT character is valid, not a data character converted by line noise to a spurious EOT. Checking for false EOT's in this manner is appropriate for many applications.

In some applications, however, the terminal continues to send data immediately after sending the EOT (as when the terminal is transmitting from a paper tape in which data is interspersed with EOT's). If the end of the receive operation in this case were delayed, the program would not recognize the EOT because of the data characters immediately following. In this instance it is necessary to specify IMEND in the FEATURE operand; IMEND causes the program to end the receive operation immediately upon detecting the EOT, without waiting to verify the presence or absence of any following characters.

Dual Code:

Either of two transmission codes can be transmitted on a communication line attached to an IBM 2701 Data Adapter Unit equipped with the dual code feature for that line. The code used is changed from one to the other by command from the access method. The same function can be performed when the IBM 3704 or 3705 is installed in place of the 2701. Specify DUALCODE in the FEATURE operand of the LINE macro representing the line if the dual code feature was used for that line when the line was attached to the 2701. Otherwise, specify NODUALCD or omit the parameter.

*Communication Line Characteristics*

As used in this book, a communication line means the entire transmission link between a station and the communications controller, including the modems (data sets), and regardless of the actual transmission medium -physical conductors (wire), microwave link, satellite link, etc., or a combination of these.

Line characteristics refer to (1) the functional attributes of the transmission path (for example, whether the line is half-duplex or duplex; (2) logical characteristics, such as the transmission code and line-control scheme employed; and (3) related aspects of the line, such as the address by which the line is known to the emulation program.

Stations may communicate with the communications controller using any of three kinds of line connections: nonswitched point-to-point, nonswitched multipoint, and switched point-to-point. (Not all types of stations can communicate with the controller over all three kinds of line connections.) You must code a LINE macro for each line connected to the communications controller, regardless of the kind of line. The LINE macro specifies to the emulation program some (but not all) of the characteristics of the line.

Whether a line is switched or nonswitched must be specified in the DIAL operand of the GROUP macro representing the line group. However, you need not specify whether a nonswitched line is multipoint or point-to-point. It is the responsibility of the host access method to be aware of this characteristic and to issue the appropriate command sequences for each type of line.

Half-duplex vs. Duplex Lines:

The emulation program must know whether a communication line is half-duplex or duplex (sometimes referred to as full-duplex). You specify this characteristic in the DUPLEX operand of the LINE macro. This operand represents the characteristics of the communication facilities, NOT the mode of data transfer over the line. Half-duplex data transfer is always used for any Start-Stop or BSC station with which the controller can communicate. It is important not to assume that a two-wire modem is necessarily a half-duplex modem; some such modems are in fact duplex. In general, if the 'clear-to-send' signal is continuously activated, the modem is duplex, regardless of whether it is a two-wire or four-wire modem. If you are in doubt, consult the supplier or installer of the modem.

Line Speeds and Clocking:

In the SPEED operand of each LINE macro, specify the data rate at which the communication line is to operate. This is the rate at which the station, controller, and modems are designed to transmit data over the communications facility that links the station and the controller.

If the modem that connects the controller has two possible data rates (as is the case with the IBM 3872 and 3875 modems, for example), designate in the DATRATE operand of the LINE macro whether the line is to operate at the higher or lower of the two rates.

In the CLOCKNG operand of the LINE macro, specify whether internal clocking (IBM business-machine clocking in the scanner) or external clocking (modem) is used for the communication line. Internal clocking is provided by the communication scanner to which the line is connected. External clocking is provided by the modem, whether the modem is a separate unit or built into the controller on the line interface bases (LIB's) 5, 6, 7, 8, and 9.

Each communication scanner in the communications controller may be provided with from one to four oscillators. The bit rate for each oscillator must be specified in the SPEED operand of the corresponding CSB macro.

Line and Subchannel Addresses:

In emulation mode, each communication line requires a nonshared subchannel address on a byte-multiplexor channel. Each line is identified to the emulation program by a line-interface address representing the physical location within the controller at which the line is attached, via a line set and line interface base (LIB). Associated with each line address are one or more CPU subchannel addresses. The multi-subchannel line-access (MSLA) facility of the emulation program permits two or more emulation subchannels to communicate, alternately, with the same communication line.

When transferring message data to or from a line, the emulation program uses the line interface address to communicate with the station and uses the associated subchannel address to communicate with the host. Specify the line interface address and the associated

subchannel address in the ADDRESS operand of the corresponding LINE macro.

*Native and Emulation Subchannel Addresses*

The native subchannel is a dedicated subchannel that must be on a byte-multiplexor channel. The host has access to the communications controller for initial program load (IPL), dump, and dynamic dump procedures through the use of this subchannel. The native subchannel address must be specified during system generation in the unit control block (UCB) of the OS, OS/VS operating system or the physical unit block (PUB) of the DOS, DOS/VS operating system.

In controlling its subchannels the emulation program operates similarly to the 2703 Transmission Control unit. The range of subchannel addresses to be recognized by the emulation program is specified in the LOCHAN and HICHAN operands of the BUILD macro. All subchannel addresses in the defined range must be dedicated to the 3704 or 3705 and must match the addresses which are physically plugged at installation.

*Modems and Automatic Calling Units*

The following information on modems and automatic calling units (ACU's) attached to the communications controller must be specified to the emulation program.

New Sync Feature:

Certain types of synchronous modems are equipped with a feature called 'new sync', which reduces the amount of line turnaround time that is normally expended each time the direction of the transmission on the line is reversed. The NEWSYNC operand on the LINE macro specifies whether this feature is to be used.

This feature is available for BSC emulation lines when:

1. The line is a duplex multipoint line and the communications controller is the master station for the line.
2. The line is a duplex, nonswitched point-to-point line and the remote station, by energizing and de-energizing its 'request-to-send' signal to the station's modem as appropriate, controls the transmission of that modem's carrier signal.

This feature is not used with:

1. Line sets 8A, 8B, or 9A.
2. Duplex lines on which the terminal carrier is not controlled by 'request-to-send'.

Determine from your IBM representative or the installer or supplier of the modem (if other than an IBM modem) whether the appropriate conditions above prevail. If not, the new sync function cannot be used and you must specify NEWSYNC=NO on the LINE macro.

Ring Indicator Mode

(Not applicable in the U.S. and Canada): Certain European modems may require their 'ring-indicator' signal line be energized (signifying the modem is being called by a station) before the communications controller indicates its readiness to receive by energizing the modem's 'data terminal ready' signal line. Signal lines constitute the interface between the communications controller and the modem.) If this requirement applies for a modem in your network, code RING=YES in the LINE macro for the communications line attached to the modem. Most modems do not have this requirement, and for these you specify RING=NO. A modem that does not have this requirement can result in an unnecessary delay in establishing the connection.

Automatic Calling Units:

Any switched callout line operating in emulation mode must be equipped with an automatic calling unit (ACU). In the AUTO operand of the LINE macro representing the line for which the ACU is used, specify the line interface address to which the ACU is attached.

*Multi-subchannel Line Access Facility*

The multi-subchannel line access (MSLA) facility of the emulation program allows the program to communicate over two type 4 channel adapters concurrently. The channel adapters may both be attached to the same host processor or may be attached to separate processors. The MSLA facility further allows two or more CPU subchannels (on the same or different channels) to communicate, alternately, with the same communication line.

In operation, a command issued over one of the subchannels seizes the line for use of that subchannel and the access method operating that subchannel. The access method retains use of the line via that subchannel until it issues a Disable command, thereby releasing the line and freeing it for use by another subchannel. (Alternately, the 3705 control panel can be used to release a line from control of one subchannel in order to switch the line to another subchannel. This action is required if the access method using the line does not issue operator-controlled Disable commands.

Subchannel-to-line associations are established during program definition and can be changed only by respecifying the associations and regenerating the program.

The physical characteristics of the line (such as type of line control, line speed, etc.) remain constant regardless of which subchannel is currently using the line. The use of the line by each subchannel must be consistent with the line characteristics. Violation of this requirement will cause unpredictable results when the access method communicates with the line.

The MSLA facility can be used in the following ways:

-- Load balancing. Communication lines can be switched from one host processor to the other during high-traffic periods to balance the load on the processors.

-- CPU backup. Communication lines can be switched to a backup host processor if the original host processor, channel, or access method fails. Execution of the control program does not end, and the program need not be reloaded into the communications controller.

-- Line sharing. Two access methods in the same or different host processors can share the same communications line alternately. The same line can thus be assigned to different applications at different times of day.

The description of the ADDRESS operand of the LINE macro explains how to associate subchannels with a line.

**Defining the Communications Controller Hardware** The hardware options with which the communications controller is equipped must be identified to the emulation program. The options are:

1. The type, number, and oscillator bit rates of the communications scanners installed in the controller.
2. The interrupt priority to be used for each line attached to the scanner.

The communications controller can be equipped with from one to four communications scanners. Each scanner may provide internal clocking for bit or character service functions for up to four different speeds of lines. Each speed requires its own oscillator within the scanner (or within the multiple speed oscillator of the type 3 scanner). For each scanner, you must specify to the emulation program:

1. Type of scanner
2. Machine model in which the scanner it is installed
3. Bit rates of the oscillators with which the scanner is equipped

The operands which define these characteristics are the TYPE, MOD, and SPEED operands of the CSB macro. One CSB macro is coded for each installed communications scanner.

The emulation program is interrupted by the line interface hardware of the controller each time a bit (type 1 scanner) or byte (type 2 scanner) is sent or received over the communication line, or when the end of buffer or end of message (type 3 scanner) occurs. To avoid bit or character overrun or underrun, lines having a high data rate require character service from the program more frequently than lines having lower data rates. Each line serviced by a given communication scanner is therefore assigned an interrupt priority relative to other lines serviced by the same scanner. If all lines on the scanner have the same data rates, the priorities may be equal. If the lines have different rates, however, those with high rates should be assigned higher priority than those with lower rates.

*IBM 3704 and 3705 Control Program Generation and Utilties Guide and Reference Manual* (GC30-3008), Appendix J: Procedure for Determining Line Interrupt Priorities provides more detail on how to determine the interrupt priority scheme for a specific configuration.

## Defining Transmission Control Unit Characteristics

When the emulation program operates a communication line, transfer of data between the CPU and the line occurs in a manner similar to that provided by the IBM 2701, 2702, or 2703 being emulated. In the CU operand of the LINE macro, you specify the transmission control unit -- 2701, 2702 or 2703 -to be emulated for that line.

When defining an emulation program, there are three important operational characteristics to be specified: the type of line-control discipline to be used for each line, the terminal time-outs required, and, (for World Trade teletypewriters only) the end-of-block and end-of-transmission sequencers to be recognized by the program.

*Type of Line Control*

All types of stations with which the communications controller can communicate in emulation mode use one of two line control disciplines: binary synchronous (BSC) or start-stop (asynchronous). Each communication line attached to the controller uses either BSC or start-stop line control in emulation mode. The same line never uses both types.

The type of line control discipline used is specified in the LNCTL operand of the GROUP macro. (All lines in the group must use the same line control discipline.)

*Terminal Time-outs*

For each communication line the emulation program normally observes two time-out intervals of several seconds duration. One of these intervals is the reply time-out, which limits the amount of time the program will await a station's response to polling or response to message data sent to the station. The other interval is the text time-out, which limits the time that may elapse between receipt of successive message characters from the station after message transmission has begun. If the time-out expires before the response or the next message character is received, the program ends the read operation for that station and notifies the access method of a time-out error. These time-outs apply to each line group in the network.

By observing these two time-out intervals, the emulation program prevents a communication line from being idled indefinitely because of excessive delay in entering successive message characters at a station or because of a malfunction or power failure at the station interrupts its transmission to the communications controller.

Unless you specify different values in the REPLYTO and TEXTTO operands of the GROUP macro, the emulation program uses the default time-out intervals.

The default value is appropriate in most cases unless your system has any of the following modifications to the Terminal Control Type 1 or 2 attached to the 2703:

- RPQ E49633, which inhibits (on a line set basis) the setting of the normal 2.0 second time-out.
- RPQ 858126, which increases flexibility by varying the length of the time-outs in increments from 0 to 28 seconds.
- RPQ W21061, which inhibits the 2.0 second time-out to allow a 28.0 second time-out to occur in response to Poll or Read and Search commands.

Some applications may justify unlimited intervals -- that is, no no time-outs at all. This characteristic (indicating time-out intervals or no time-out) may also be specified in the REPLYTO or TEXTTO operands.

*EOB and EOT Sequences*

You may specify the character sequence the emulation program is to recognize when receiving from a terminal as the end-of-block (EOB) and end-of-transmission (EOT) sequences. This capability requires RPQ Y24344 (the same characters four times) and RPQ M33455 (different center character in FIGS/LTRS).

The EOB sequence may be either FIGS x or nnnn. x and n may be any telegraph code combination except a combination representing the FIGS or LTRS character. (If the terminal is equipped to send who-are-you (WRU) sequences, x also may not be the letter D.)

The EOT sequence may be FIGS y LTRS; y may be any applicable telegraph code combination except one representing FIGS, LTRS, or the same x character used in the EOB sequence, FIGS x.

Specify the required sequences in the EOB and EOT operands of the GROUP macro representing a World Trade teletypewriter line group.

**Diagnostic and Service Aids** The emulation program provides the line trace, panel test, and dynamic dump facilities to aid in diagnosing difficulties in network operation. Inclusion of these optional facilities in the program is recommended.

*Line Trace Facility*

The line trace facility of the emulation program is a service aid that permits detailed analysis of the operation of any communication line controlled by the program. This facility records operating parameters of a line each time a level 2 (except bit service) or level 3 interrupt occurs for that line. (Level 2 is the program level at which bit service, character service, or type 3 scanner buffer service for the communication line is initiated. Level 3 is the program level at which servicing of channel interrupts in performed.) The program accumulates this information in a trace table within controller storage. The contents of the controller storage must be dumped (using the dump or dynamic dump utility) in order to make the line trace records available. Inclusion of the line trace facility, the number of lines to be traced, and the size of the trace table are specified in the LINETRC operand of the BUILD macro.

The line trace facility does not interfere with normal operation of the communication line. Performance may diminish somewhat because of the additional processing needed each time a character service interrupt occurs for the line or lines being traced. The amount of decrease in performance depends upon how heavily the communications controller is currently loaded. Inclusion of the line trace facility has no effect on performance except when a line is actually being traced.

Emulation line traces are initiated at the control panel of the communications controller. Any number of lines may be traced concurrently in an emulation program. Only one line at a time may be traced in the network control program.

*Panel Tests*

Certain tests of communication lines can be run from the control panel of the communications controller. The panel-initiated line tests can be run only if the test

function is included in the emulation program by specifying TEST=YES on the BUILD macro. Using these test routines, the operator at the controller can perform many of the telecommunication functions (such as polling, addressing, and data transfer) normally executed by the controller and its control program upon request from the access method.

*Dynamic Dump Facility*

The dynamic dump facility is a service aid that transmits communications controller storage contents to the host processor without stopping the execution of the emulation program. You can obtain a full storage dump or a dump of the trace table. Additionally, the line trace can be activated or deactivated from the control panel. Portions of controller storage can also be displayed on the operator's console at the host processor.

The DYNADMP operand of the BUILD macro specifies whether the dynamic dump option is to be included in the emulation program.

**Program Generation Options and Data Sets** All the options described thus far have related to the operational characteristics of the telecommunication subsystem. This section describes several options affecting the generation procedure and the emulation program data sets (files) used in the procedure.

*Program Generation Options*

Program generation options pertain to the type of functions the emulation program is to perform, the type of communications controller in which the program will be executed (3704 or 3705), and several assembly and link-editing options. All program generation options are specified in the BUILD macro.

Model of Controller:

The emulation program can be executed in either an IBM 3704, 3705 I, or 3705 II Communications Controller. Minor internal differences between the three controllers require that you specify, in the MODEL operand of the BUILD macro, the type of controller in which the emulation program is to be loaded and executed. Changing the value in this operand is the only modification required to allow an emulation program originally defined for one type of controller to be executed in the other type, provided the subsystem configurations are identical. That is, the network configuration (including line and subchannel address assignments), the controller configuration (number and type of channel adapters and communication scanner, and storage size), and operational options must be the same for both controllers.

Other Options:

The remaining program generation options, and the operands of the BUILD macro by which you specify these options are:

-- The name to be given to the generated program load module. This name is defined in the NEWNAME operand of the BUILD macro.

-- (OS, OS/VS only) Whether stage 2 of the generation procedure is to consist of a single, multistep job or a separate job for each step, and whether a job card is required. These options are defined on the JOBCARD operand of the BUILD macro. The JOBCARD operand is valid for DOS/VS NCP generation.

-- (OS, OS/VS only) The type of device or class of devices to be used for utility data sets during stage 2 generation. The class of device is defined on the UNIT operand of the BUILD macro.

*Data Sets Used in the Generation Procedure*

The following operands apply only to OS or OS/VS systems.

The names of the various program data sets to be used in the generation procedure when generating under OS or OS/VS are specified by the LOADLIB, OBJLIB, QUALIFY, UT1, UT2, and UT3 operands of the BUILD macro.

### Storage Requirements

*Storage Estimates for the IBM 3704 and 3705 Communications Controllers Emulation Program* (GC30-3005) provides calculations on storage requirements.

### Defintion Review

The emulation program is defined by five IBM-supplied macros. The system macros (BUILD and CSB) provide information pertaining to the entire controller, such as hardware features, certain control program options, and program generation information. The configuration macros (GROUP and LINE), defining the specific communications network, provide information necessary to construct the tables needed by the emulation program to control the flow of data between the controller and the telecommunication network, and between the host processor and the communications controller. The generation delimiter macro (GENEND) ends the control program generation input stream.

The macro sequence is as follows:

BUILD-- First macro of a generation

CSB-- One per communications scanner. All CSB macros follow the BUILD macro and are coded before the first GROUP macro

GROUP-- One to 255 line groups

LINE-- One to 32 lines (3704), one to 352 lines (3705). One or more LINE macros follow each GROUP macro.

GENEND-- Last macro of a generation

The operands are all nonpositional keyword operands. Operands of the LINE macros may be 'promoted' to the GROUP macro level, which allows an operand to be coded once per GROUP. The operand then applies to all LINE macros of the GROUP. A LINE macro operand coded at the GROUP level can be overridden by coding the same operand at the LINE level. This method allows an operand applying to a majority of the lines to be coded at the GROUP level, then reversed for a specific line or lines.

### Definition Quiz

The following questions allow you to evaluate yourself on the previous reading material. The answers are given in Appendix A.

Multiple choice:

1. The sequence of emulation macros for an emulation generation is:

   a.
   ```
   BUILD
   CSB
   GROUP
   LINE
   GENEND
   ```

   b.
   ```
   CSB
   GROUP
   LINE
   BUILD
   GENEND
   ```

   c.
   ```
   GROUP
   LINE
   BUILD
   CSB
   GENEND
   ```

   d.
   ```
   BUILD
   GROUP
   LINE
   CSB
   GENEND
   ```

True or False:

2. Operands are not positional.
3. Some operands of the LINE macro may be coded on the GROUP macro.
4. Some operands of the GROUP macro may be coded on the BUILD macro.
5. Line trace, panel tests, and dynamic dump are optional facilities selected by operands.

## BUILD Macro

### Objective

Upon completion of this topic, the student should be able to select and code the appropriate operands of the BUILD macro for a defined emulation configuration.

### BUILD Introduction

The previous section gave you an overview of emulation specification. This section provides the information on the first macro of an emulation generation. This macro has different operands for DOS-DOS/VS and OS-OS/VS. At the end of this section, you will be asked to code the macro and operands for a defined configuration. Following each macro coding problem is a review of the correct operands. You should not look at the review until you have coded all of the operands of the macro problem. This method will allow you to identify those operands you should review.

You will need the following reference manual to code the emulation generation problems:

*IBM 3704 and 3705 .Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008), Chapter 7: EP Generation Macro Instructions for BSC and/or Start-Stop Networks -- System Definition Macro Instruction. The reference for both DOS and OS nonVS emulation program is *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3002).

Read the BUILD macro instruction and operand references, but do not read the Configuration Definition Macro Instructions (CSB, GROUP, and LINE) which follow the BUILD macro until you have coded the BUILD macro problem.

### BUILD Macro Review

The BUILD macro is the first macro in a series of macros required for an emulation program. BUILD is the system definition macro, and as such defines the attributes of the communications controller, the options available in the emulation program, and the type of operating system.

The BUILD macro requires certain operands, which are different for OS/VS and DOS/VS. Figure 2.2 provides a summary of the operands for each system.

**OS/VS**

| | |
|---|---|
| Required operands | HICHAN= |
| | LOCHAN= |
| | LOADLIB= |
| | OBJLIB= |
| Optional operands | CA= |
| | DYNADMP= |
| | JOBCARD= |
| | LESIZE= |
| | LINETRC= |
| | MODEL= |
| | NEWNAME= |
| | OPCSB2= |
| | QUALIFY= |
| | TEST= |
| | TYPGEN= |
| | TYPSYS= |
| | UNIT= |
| | UT1= |
| | UT2= |
| | UT3= |

**DOS/VS**

| | |
|---|---|
| Required operands | HICHAN= |
| | LOCHAN= |
| | TYPSYS= |
| Optional operands | CA= |
| | DYNADMP= |
| | LINETRC= |
| | MODEL= |
| | NEWNAME= |
| | OPCSB2= |
| | TEST= |
| | TYPGEN= |

**Figure 2.2 BUILD macro operands.**

### BUILD Macro Quiz

Code the appropriate BUILD macro operands for the following configuration:

You are testing your understanding of the material by solving this problem. Be sure to finish the problem before referring to the solution in Appendix A.

A 3704 on a VS system has a Type 1 Channel Adapter with eight subchannel addresses starting at subchannel X'10'. The native subchannel address (IPL address) is subchannel X'17'. The four-byte buffer is to be used rather than the twenty-byte buffer. All optional trace and test facilities are to be included in the emulation generation.

*OS-OS/VS Only*

The data sets required for OS-OS/VS all have a SYS1 qualifier. The object library is OBJ3704. The load library is LOAD3704. The member name is TEST1. All other job control definitions are optional.

Criterion

If you make more than one coding error while coding the BUILD macro operands, you should review this section, in order to avoid difficulty with the BUILD macro in the future.

## CSB Macro

### Objective

Upon completion of this topic, the student should be able to select and code the appropriate operands of the CSB macro for a defined emulation configuration.

### CSB Macro Introduction

This section provides the information on the CSB macros required for an emulation generation. This macro is required for each scanner of a 3704 or 3705. For a single scanner, only one CSB macro is required. A 3705 with four scanners requires four CSB macros. All CSB macros must follow the BUILD macro and precede all other emulation macros.

The following material is needed for coding the problem.

*IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008), Chapter 7: EP Generation Macro Instructions for BSC and/or Start-Stop Networks -- CSB Configuration Definition Macro Instruction.

Read the section on the CSB macro. Do not read the section on the GROUP, LINE, and GENEND macros until you have coded the CSB macro problem.

### CSB Macro Review

The CSB macro, like the BUILD macro, has certain operands that are required. Two common errors to avoid in writing CSB macros are: (1) not identifying all clocks, and (2) not specifying clocking rates from the lowest to the highest. Clocks are always installed from the lowest rate to the highest and should be coded in the same sequence.

Figure 2.3 illustrates the required operands of the CSB macro.

```
CSB   SPEED=(rate1,rate2,rate3,rate4),
      TYPE=scanner-type,
      MOD=0|n,
      WRAP=lineaddr
```

**Figure 2.3 CSB macro operands.**

### CSB Macro Quiz

Code the appropriate CSB macro for the following configuration:

The installation has a type 2 communications scanner with a clock of 134.5 rate. The wrap line for diagnostic tests is at the lowest line address of the scanner.

It is advisable not to consult our solution until you have coded all of the operands. After you have completed coding the CSB Macro, refer to the solution in Appendix A.

Criterion

If you make more than one coding error while coding the CSB macro operands, you should review this section in order to avoid difficulty in this area in the future.

## GROUP Macro

### Objective

Upon completion of this topic, the student should be able to select and code the appropriate operands of the GROUP macro for a defined emulation program.

**GROUP Macro Introduction** The previous macros defined the generation information and the communications controller hardware. GROUP is the first macro to define the characteristics of the communication lines. The GROUP macros follow BUILD and all CSB macros; however, the GROUP macros are separated by one or more LINE macros. After the reading assignment, you will be asked to code the GROUP macros for a defined emulation configuration. Code each GROUP macro on a separate coding sheet. The LINE macros are coded following the appropriate GROUP macro.

You will need the following material for coding the GROUP macro problem:

*IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008), Chapter 7: EP Generation Macro Instructions for BSC and/or Start-Stop Networks -- Network Configuration Macro Instructions -- GROUP Macro Instruction

Read the material on the GROUP macro. Do not read the material on the LINE and GENEND macros until you have coded the GROUP macros for the problem.

**GROUP Macro Review** In the GROUP macro, it is possible to omit all the operands. If the operands are omitted, a nonswitched, start/stop line with standard timeouts is assumed. The GROUP macro follows BUILD and all CSB macros. Each GROUP macro is followed by one or more LINE macros.

**GROUP Macro Quiz** Code the appropriate GROUP macros for the following configuration:

There are two BSC lines and two SS lines. The BSC lines are nonswitched. The SS lines are switched, dialin only lines. Standard timeouts are to be defined. BSC lines are for IBM BSC 3270s; SS lines are for IBM 2741s.

Code each GROUP macro on a separate page so that LINE macros may be added following each GROUP macro. After you have finished coding the GROUP macros of the problem, refer to the solution in Appendix A.

Criterion

If you make more than one coding error while coding the GROUP macro operands, you should review this section in order not to experience difficulty in this area in the future.

## LINE Macro

### Objective

Upon completion of this topic, the student should be able to select and code the appropriate operands of the LINE macro for a defined emulation configuration.

**LINE Macro Introduction** The previous macros defined the generation information, the communications controller hardware, and the line groups. The next task is to define the lines within the line group definitions.

Each communications path must be defined by a LINE macro. The LINE macros must follow the appropriate GROUP macro; however, there is no required sequence of LINE macros following a GROUP macro. Many operands of the LINE macro can be moved up and coded as operands of the GROUP macro, rather than specified on the LINE macro. Operands 'promoted' to the GROUP then apply to all LINE macros which follow the GROUP macro (unless overridden by the operand coded on a specific LINE macro).

After the reading assignment, you will be asked to code the LINE macros and operands for a defined emulation configuration. The LINE macros should be coded following the appropriate GROUP macro.

The following material is needed for your reading assignment and for coding the LINE macro problem:

*IBM 3704 and 3705 Communications Controllers Control Program Generation and Utilities Guide and Reference Manual* GC30-3008), Chapter 7: EP Generation Macro Instructions for BSC and/or Start-Stop Networks -- Network Configuration Macro Instructions -- LINE Macro Instruction

Read the material on the LINE macro. Do not read the material on the GENEND macro until you have coded the LINE macros for the problem.

Read the following topics:

*Teleprocessing Preinstallation Guide for IBM 3704 and 3705 Communication Controllers* (GC30-3020),
(1) Parameter Selection Tables: Stations and
(2) Parameter Selection Tables: Line Sets

**LINE Macro Review** The reading assignment has defined a LINE macro instruction and the operands of the LINE macro. Most of the operands are self-explanatory; however, there are certain items that are worthy of emphasis.

In Figure 2.4, the operands listed within the dotted lines can be used in either the GROUP or LINE macros, whereas those listed under LINE can be used only in LINE macros and those listed under GROUP can be used only in GROUP macros.

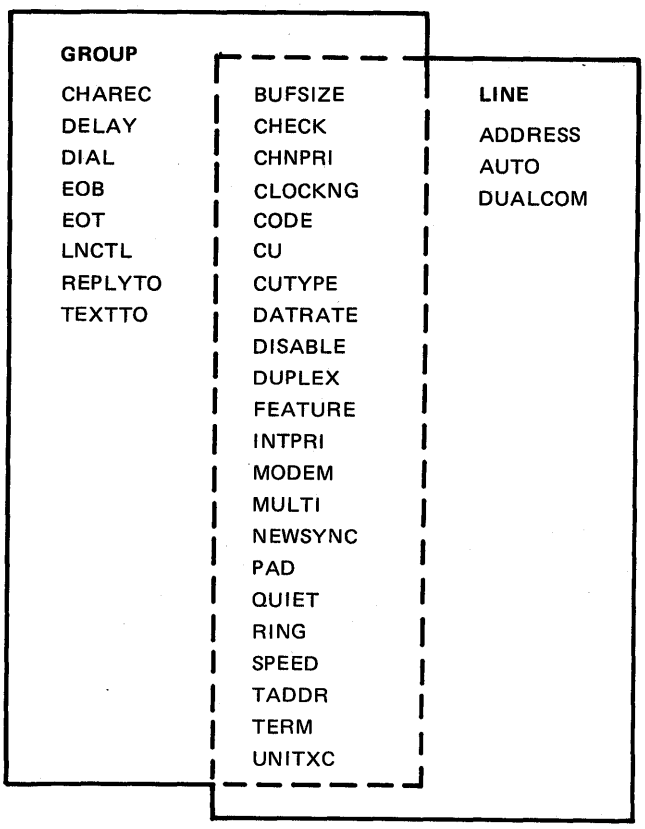



**Figure 2.4 GROUP and LINE macro operands.**

When coding the ADDRESS operand, be careful in specifying your subchannel value. A common error is to specify more than two digits for the subchannel identification.

**LINE Macro Quiz** Code the appropriate LINE macros for the following configuration:

The type 2 communications scanner has four lines at the first four scanner address positions. The first two lines are nonswitched, BSC lines of EBCDIC 3271 control units with 3277 terminals. Note: When the CUTYPE operand is defined the TERM operand is ignored. The subchannels defined for the BSC lines are 10 and 11. Line priority interrupts should be 1 for BSC lines.

The third and fourth lines are switched, callin, start-stop lines for IBM 2741 terminals which do not have LRC checking or immediate end feature requirements. Line priority interrupts should be 0 for start-stop lines. An EOT received on a start-stop line should signal unit exception to the host. The wrap line for diagnostics is the first scanner address.

All four lines have single speed modems which do not require a 'long disable timeout'. The modems are disabled until enabled by command from the access method (the 'data set ready' lead of the modem is not continuously energized). The interface connection between the line sets and modems (modem strapping) is full duplex. The BSC modems provide clocking at 2400 bps. The start-stop lines use internal clocking at 134.5 bps. Data carrier detect monitoring is not required. The BSC lines have the 'new sync' option. The controller is to verify that the first four bits of trailing pad are all '1's.

Code the LINE macros following the appropriate GROUP macro. Promotable operands may be coded on either GROUP or LINE macro.

After you have finished coding the appropriate LINE macros, each following the appropriate GROUP macro, refer to the solution in Appendix A.

Criterion

If you make more than three coding errors when coding the LINE operands, you should review this section, so as to avoid difficulty in this area in the future. The introduction to the previous section, Defining the Emulation Program, provided information on the general concepts for operands coded on the LINE macro. This section should also be considered for additional review or reference.

## GENEND Macro

### Objective

Upon completion of this section, the student should be able to select and code the appropriate operands of the GENEND macro for a defined emulation program.

**GENEND Macro Introduction** The previous sections provided the definition of the telecommunications network. This section explains the last macro of an emulation generation. This delimiter macro gives the user control of the scanners. You can provide an upper scan address limit for each type 2 or type 3 communications scanner, which enables the scanner to avoid looking at interface addresses that do not have lines attached (or defined), and thus to service existing lines more frequently for faster line service. You can also specify that by address substitution up to four lines can be scanned each eight addresses.

The following reference is needed for your reading assignment and to code the GENEND macro.

*IBM 3704 and 3705 Communications Controllers Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008), Chapter 7: EP Generation Macro Instructions for BSC and/or Start-Stop Networks -- GENEND Macro.

Read the material on the GENEND macro.

**GENEND Macro Review** The GENEND macro completes the first stage of the generation procedure. GENEND provides the definition for upper scan limit and address substitution value for type 2 and type 3 scanners.

For type 2 scanners, if you have one or more lines greater than 4800 bps, you must code the scan control for either an upper scan limit or address substitution. If you set the value to upper scan limit, you affect only one scanner; however, if you use address substitution, you affect the addresses of all scanners.

For type 3 scanners, for high-speed lines you should use the 'high speed select register' rather than upper scan limit. Because address substitution is controlled by the attachment base, all type 2 and type 3 scanner addresses are affected.

**GENEND Macro Quiz** Code the GENEND macro and operands as the last macro of the emulation job stream. Address substitution and upper scan limit are not required.

After you have coded the GENEND Macro, refer to the solution in Appendix A.

Criterion

If you make one mistake in this problem, it is recommended that you review this section. Also review upper scan limit, address substitution, and high speed select in the *IBM 3704 and 3705 Communiations Controller Hardware* (SR20-4544).

# System Support Programs

### Objective

Upon completion of this section, the student should be able to provide the job control required to generate an emulation program.

### Generation Procedures

In the previous sections, you have written the macros and operands to generate an emulation program for a defined telecommunications network. You have the macros which now are ready to be assembled to generate the emulation program.

In the following reading assignment, there is a section for DOS-DOS/VS and a separate section for OS-OS/VS. Please select the reading assignment based upon your requirements and background.

### Emulation Program Generation Under OS/VS

The emulation program generation procedure is a two-stage process consisting of a series of jobs executed under the control of the host operating system. You must code the entire stage 1 input job stream. The stage 2 job stream does not require coding as it is produced automatically by stage 1.

Stage 1 of the generation procedure is an assembly using either the communications controller assembler or a host (OS/VS only) assembler. The emulation program generation macros (describing the emulation program to be generated) are the input to the assembly; the output is a job stream (sequential data set) which is the input to stage 2. This job stream can be directed to cards, tape, or a direct access storage unit. The stage 1 output (stage 2 input) contains (1) data constants, (2) macros that will cause stage 2 to generate control tables and conditionally assemble the required program modules, (3) job control statements for stage 2, and (4) linkage editor control statements.

The 3704 and 3705 assembler program name for OS/VS is CWAX00. The OS name is IFKASM. The data set requirements are identical to the host assembler.

Operator intervention is required between the two stages of program generation. Diagnostic messages produced at the end of stage 1 indicate any errors that have occurred.

Diagnostic messages for emulation macros for OS/VS are given in *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008), Appendix B: Network Control Program Generation Messages. Diagnostic messages for emulation macros for OS nonVS systems are given in *IBM 3704 and 3705 Emulation Program Generation and Utilities Guide and Reference Manual* (GC30-3002), Appendix D: Emulation Program Generation Messages. Diagnostic messages for the 3704 and 3705 assembler are given in *IBM 3704 and 3705 Communications Controllers Assembler Language.*

Serious errors during generation result in an incomplete stage 1 job stream. The source statements must be corrected and stage 1 reexecuted. If no serious errors occur in stage 1, the operator initiates the second stage, specifying as input the stage 1 output.

Stage 2 of the generation procedure uses the communications controller assembler to assemble conditionally the control tables (CYALNVT, CYACHVT, CYAEPCCB, and CYAEPLGT) required by the emulation program you are generating and places them on the library you specify in the OBJLIB operand of the BUILD macro. The linkage editor then combines the preassembled modules (from the IBM-supplied modules) and the control tables (from the OBJLIB data set) into an emulation program load module and places this load module in the library you specify in the LOADLIB operand of the BUILD macro.

In addition to the load module produced by the linkage editor, unresolved external references are also produced. Each item not included in the emulation generation creates an unresolved address for the omitted resource. If you defined a type 2 scanner, address constants that refer to a type 1 scanner will be unresolved.

Upon the completion of stage 2, the load module is ready to be loaded into the communications controller from the host processor.

Note: Multiple emulation program load modules must have different names. Otherwise, unless a different library is specified, the latest load module will replace the previous copy.

Figure 3.1 illustrates the contents of the stage 1 input job stream using the host (or 3704/3705 Assembler) and the output of stage 1 used as input to stage 2 to create the emulation load module.

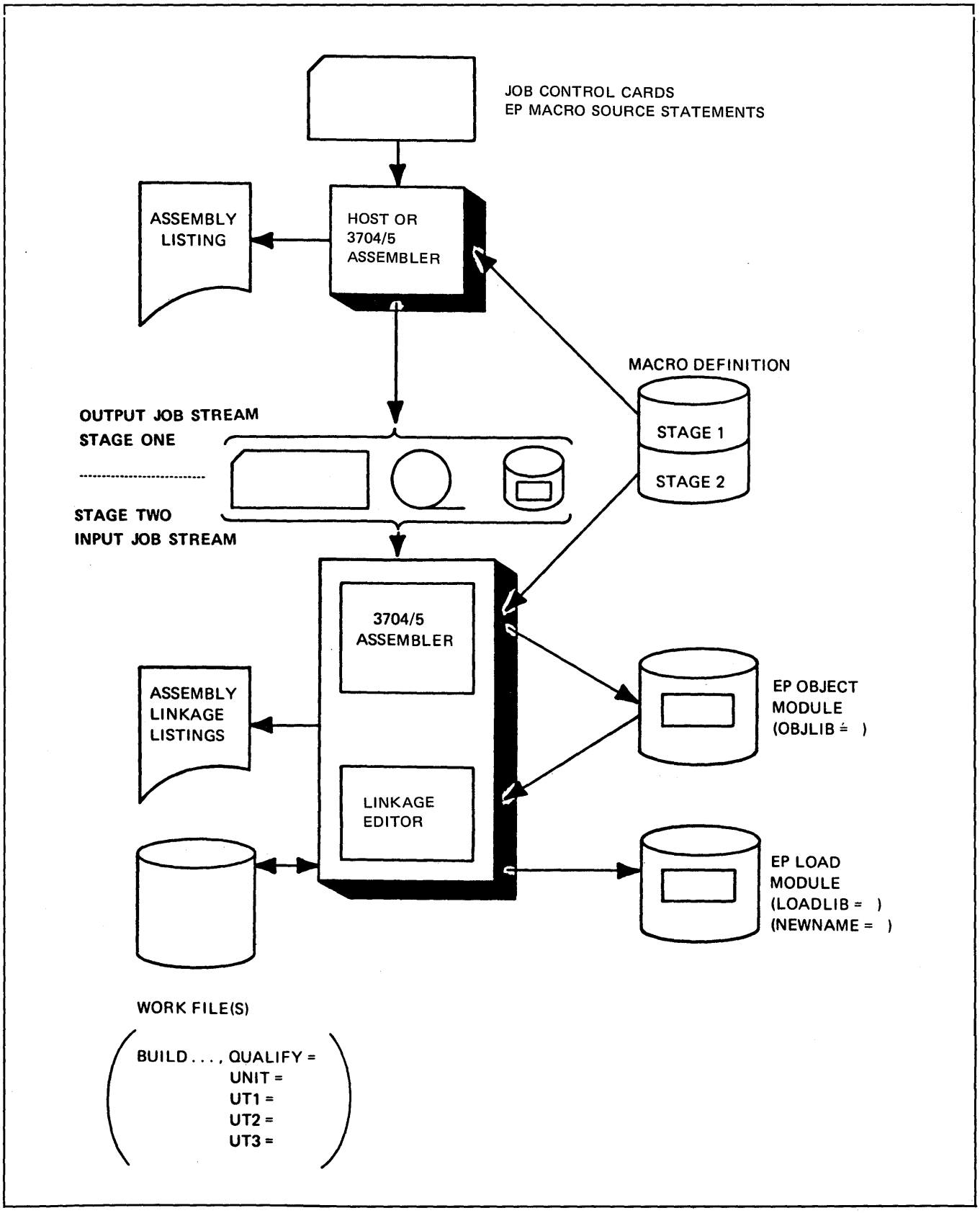



Figure 3.1 Emulation generation procedure for OS-OS/VS.

### Emulation Program Generation Under DOS/VS

The emulation program generation procedure under the control of the disk operating system (DOS/VS) is a two-stage process.

In the first stage of the generation procedure, the macros you have coded are assembled by the communications assembler. The communications assembler is required because the first stage creates a type of address constant which is not valid for a host assembler. The output from the assembler is an object deck and a printed listing. The stage 1 job stream and emulation macros are illustrated in Figure 3.2. The printed output lists the generated object deck code and also lists linkage editor INCLUDE cards identifying the IBM-supplied modules which are required for the emulation program load module.

The DOS/VS program name of the 3704 and 3705 assembler is IFZASM. The DOS 3704/3705 assembler is IFTASM. The file requirements are the same as for the host assembler.

Operator intervention is required between the two stages of program generation. Diagnostic messages are printed (stage 1 output listing) at the end of stage 1 indicating any errors that may have occurred.

Diagnostic messages for emulation macros for DOS/VS are in *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008), Appendix B: Network Control Program Generation Messages. Diagnostic messages for emulation macros for DOS/nonVS are in *IBM 3704 and 3705 Emulation Program Generation and Utilities Guide and Reference Manual* (GC30-3002), Appendix D: Emulation Program Generation Messages. Diagnostic messages for the 3704 and 3705 assembler are given in *IBM 3704 and 3705 Communications Controllers Assembler Language.*

If the generation results in serious errors, no object deck is produced. The source statements must be corrected and stage 1 reexecuted. If no serious errors occur in stage 1, the operator creates the second stage.

Stage 2 of the generation procedure is a two-step process. In the first step, the object module produced by stage 1 contains a generated CATALR card which is used to catalog this object module in a relocatable library, using the MAINT utility program. In the second step, you code the link-edit step that will join the stage one object module with the preassembled emulation program load modules specified in the stage 1 output listing. you must punch the INCLUDE cards in the exact order listed on the output from stage 1 (the cards are listed just before the END statement) and include the cards in that sequence in the linkage-edit step. The result is an emulation program load module that is located in a core image library.

After the link-edit step (when the emulation program load module is in the core image library), the user-coded CSERV utility program must be executed to move the load module to a sequential file that you specify in your DLBL and EXTENT cards, before the loader utility program can load your emulation program. The CSERV utility requires that the host system allow assignment of SYSPCH to a direct access file. This capability must be included in your supervisor by adding the SYSFIL operand in the FOPT macro when assembling the supervisor.

In addition to the load module produced by the linkage editor, unresolved external references also are produced. You will have an unresolved reference for type 1 scanner code if you specified a type 2 scanner for your generation, and each optional support omitted produces a similar unresolved external reference.

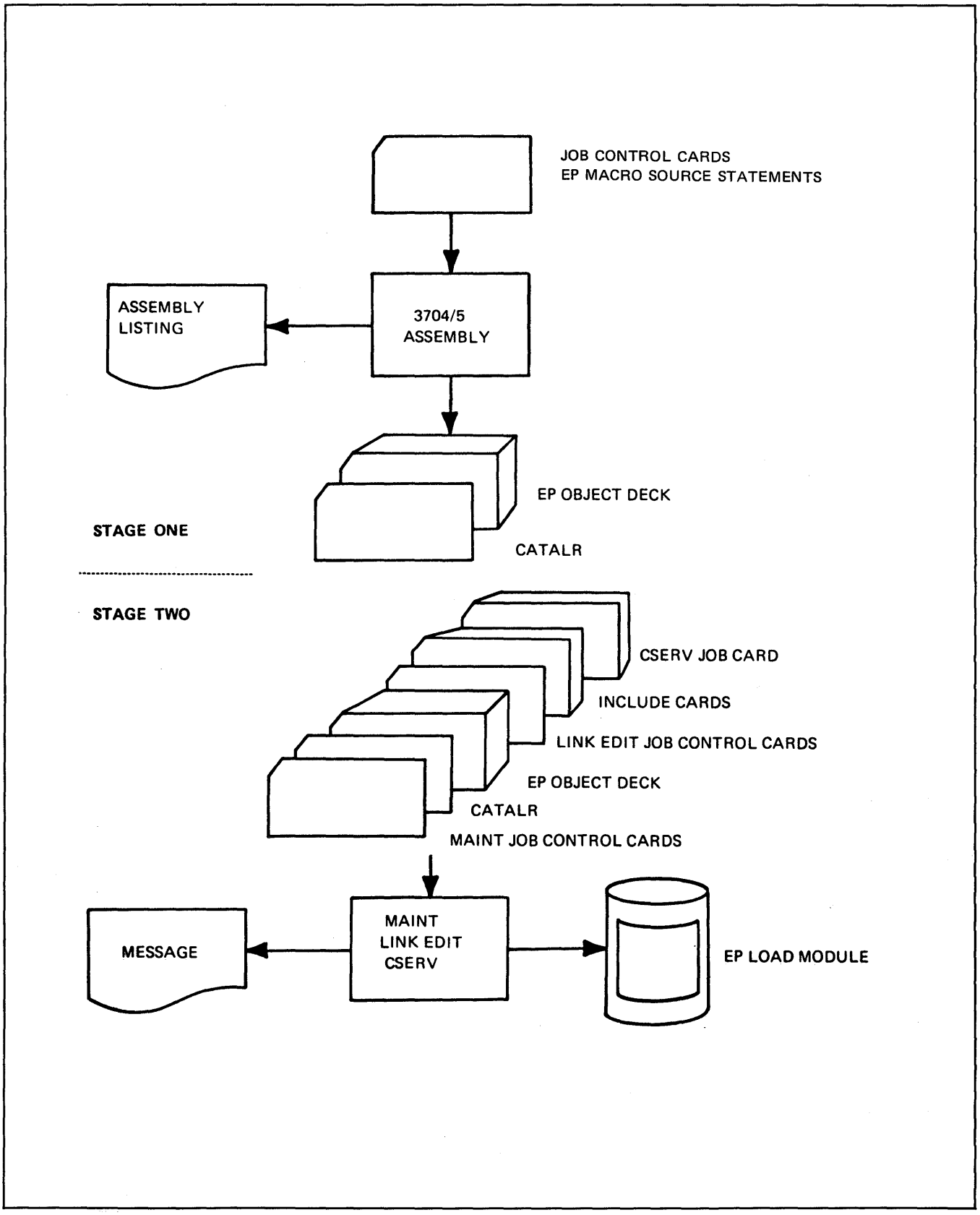



Figure 3.2 Emulation generation procedure for DOS-DOS/VS.

**Generation Review** Generation of an emulation program is a two-stage sequence. The emulation generation requires an assembly in stage 1 of emulation macros. The output of stage 1 is used as input to stage 2, which creates an emulation load module.

Keep in mind that there will be unresolved external references in the linkage editor output when:

1. The trace option is not specified.
2. The panel test option is not specified.
3. Start/stop terminals are specified, but no display terminals are specified.
4. No start/stop terminals are included.
5. No BSC terminals are included.
6. The dynamic dump option is not specified.
7. The DELAY operand is not specified on the GROUP macro.
8. No TTY or TWX terminals are included.

**Generation Quiz** Answer the following questions, referring to the reading material as required. Do not refer to the answers until you have answered all of the questions. The solution is in Appendix A.

1. What is the standard job control program name of the 3704 and 3705 communications controller assembler for your operating system?
2. What is the form number of the SRL which contains the assembler diagnostic messages (not macro diagnostic messages)?
3. What data sets are required for the 3704/3705 assembler?
4. What is the output of the stage 1 emulation generation assembly?

Criterion

If you missed more than one question, you should review this section.

## Loader Utility

### Objective

Upon completion of this topic, the student should be able to identify the job control cards and code the utility control card and operands to load an emulation program.

**Loader Utility Introduction** In the previous section you read about the job control to generate your emulation program. This section explains how to load the generated program into the communications controller. At the completion of the reading assignment, you will be asked to identify the job control cards and code the loader utility control card to load the emulation program.

The loader utility program transfers an emulation program load module from the host processor to the communications controller. The loader utility must be run as a job or job step in the host system.

The loader has two modules. One is an operating system utility that may be invoked like any other host utility. The other module runs in the communications controller. When the loader is invoked, the controller module is contained within a data area in the host processor loader module. The host processor module loads the controller module into the controller via an initial program load (IPL) command.

The communications controller module of the loader can be executed in any communications controller. The only requirements for the load operation are that the communications controller be identified to the operating system (and generated in the operating system), that it be free to be allocated to the loader job step, and that its power be on.

Before loading the emulation program into the controller, the loader utility may optionally load a diagnostic routine called the initial test routine. This routine tests the communications controller for hardware malfunctions that might later cause failure of the emulation program and sets good parity in emulation program storage areas. If the initial test routine detects no malfunctions, the loader then loads the emulation program into the controller. If the initial test routine does detect trouble, the routine stops and the loader issues an error message to the host console.

Diagnostic messages for OS/VS are in *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008), Appendix C: Utility Messages. Diagnostic messages for OS nonVS and DOS/VS systems are in *IBM 3704 and 3705 Emulation Program Generation and Utilities Guide and Reference Manual* (GC30-3002), Appendix E: OS Utility Messages; Appendix F: DOS Utility Messages.

Loading the initial test routine is optional (the routine is run unless you specify its omission in the LOAD control statement), but the test is recommended because it can detect conditions that later may cause failure of the emulation program. Running the initial test routine is especially recommended for a communications controller which has just been powered on because the test sets good parity in storage.

Successful completion of the emulation program loading process is indicated to the host operator by a console message.

The material that follows discusses operating system dependencies by operating system. Please select the appropriate reading for your interest and background.

**Loader Utility for OS/VS** The program name of the loader utility is IFLOADRN. IFLOADRN is the name used on the EXEC card of the job which invokes the utility. Input to the OS/VS loader utility is as follows:

1. A DASD partitioned data set containing the emulation program load module to be loaded. This partitioned data set is identified by SYSUT1. The member name is provided by the loader utility control card operand of LOADMOD=name.
2. The SYSIN data set contains the LOAD control statement, which specifies the name of the emulation program load module and the communications controller into which the module is to be loaded.
3. The SYSUT3 DD card identifies a partitioned data set containing the initial test routine. The initital test routine consists of two load modules, IFL3705D and IFL3705E, that are to be loaded before emulation program loading. This data set is optional and may be omitted if the initial test routine is not desired (as indicated by DIAG=NO in the LOAD statement).

The loader consists of the load modules IFLOADRN, IFLLD1P2, and IFLLD2P2. These modules must be in SYS1.LINKLIB data set or on a partitioned data set pointed to by a STEPLIB or JOBLIB statement. If the initial test routine is desired, the loader must also consist of IFL3705A and IFL3705B.

A DD card is required to define the address of the 3704 or 3705 controller to be loaded. The DD label must match the definition in the load utility control card operand of 3705=name.

An additional output from the loader program is the SYSPRINT data set. This data set contains the completion messages and/or error messages produced by the loader.

The OS/VS job control statements needed to invoke the loader program are specified on Figure 3.3

```
//jobname    JOB
//stepname   EXEC   PGM=IFLOADRN
//SYSUT1     DD     Partitioned data set of emulation load
                    module.
//SYSUT3     DD     Partitioned data set of initial test routine.
//SYSPRINT   DD     Sequential message data set.
//ddname     DD     Unit=address of controller to be loaded.
//SYSIN      DD     Source of utility control statement.
```

**Figure 3.3 Load utility job control statements.**

There is one utility control statement, the LOAD statement. This statement may not be omitted and must be contained in one 80-character card image (the loader program does not recognize continuation characters). The LOAD statement specifies:

1. Which member of the SYSUT1 defined input data set identifies the emulation program load module to be loaded (LOADMOD=member).
2. Which DD card identifies the communications controller is to be loaded (3705=name).
3. Whether or not the diagnostic initial test routine is to be executed before the emulation program load module is loaded (DIAG=NO | Y6 | Y8). An operand of NO bypasses the initial test, Y6 performs initial test for a machine of 64K or less storage, and Y8 for storage over 64K.

The format of the loader control card is as follows, beginning in column two or later:

```
LOAD  LOADMOD=member,
      3705=ddname,
      DIAG=Y6|Y8|NO
```

**Loader Utility for DOS/VS** The program name of the DOS/VS loader utility is IFULOAD. Input to the DOS/VS loader utility is as follows:

1. A sequential DASD file containing the emulation program load module to be loaded. The DLBL filename operand (first DLBL operand) is identified in the LOADMOD operand of the utility control card.
2. A SYSIN file containing the control statement LOAD.
3. A file containing the initial test routine (consisting of modules IFU3705D and IFU3705E) to be loaded before emulation program loading. This file is optional; it may be omitted if the initial test routine is not desired (as indicated by DIAG=NO in the LOAD statement). This DLBL filename must be DIAGFLE.
4. If the initial test routine is desired (as indicated by DIAG=Y6 or Y8), the loader utility requires that interval TIMER support be present and assigned to the background. Before loading the communications controller for the first time, two initial test modules must be moved to a direct access file to which the loader has access. The CSERV utility program must be used to create this file. The DIAG=NO operand can be coded with one of three values:

DIAG=NO-- Bypasses the initial test routine

DIAG=Y6-- Tests for controller addressability of 64K or less

DIAG=Y8-- Tests for controller addressability of more than 64K

The DOS/VS job control statements needed to invoke the loader program are illustrated on Figure 3.4.

```
// JOB
// ASSGN    SYSxxx     Identifies the controller to be loaded.
                       The SYSxxx must match the 3705=
                       SYSxxx operand of the LOAD
                       statement.
// DLBL     filename   Filename must match the LOADMOD=
// EXTENT   SYSyyy     filename.
// ASSGN    SYSyyy
// DLBL     DIAGFLE    The initial test routine.
// EXTENT   SYSzzz,
// ASSGN    SYSzzz
// EXEC     IFULOAD
```

**Figure 3.4 Loader utility job control for DOS-DOS/VS.**

There is one utility control statement, the LOAD statement. This statement may not be omitted and must be contained in one 80-character card image (the loader program does not recognize continuation characters). The LOAD statement specifies:

```
LOAD LOADMOD=filename,
     3705=SYSxxx,
     DIAG=Y6|Y8|NO,
     DEVICE=dasdtype
```

The LOADMOD operand must match the DLBL filename defining the sequential emulation program to be loaded. 3705=SYSxxx defines the SYS assigned to the controller to be loaded. DIAG operand can be coded NO for no initial test routine, Y6 for controllers with 64K storage or less, or Y8 for controller storage of over 64K. The DEVICE operand specifies the DASD type containing the emulation program; 2311, 2314, 3330, or 3340.

**Loader Utility Review** The loader utility program name for OS/VS is IFLOADRN; DOS/VS is IFU-LOAD. The loader requires definition of data sets (files) identifying the 3704 or 3705 to be loaded, the source of the emulation program, the initial test routine (optional), an 80-character utility control card, and a print file for messages.

The load utility requires a LOAD statement, the control statement for the communications controller being loaded. This statement, which may not be omitted, specifies:

1. The name of the data set that contains the emulation program load module to be loaded
2. The symbolic address (DOS) or DD label (OS) of the communications controller to be loaded
3. An indication of whether the diagnostic routine should be executed
4. (DOS only) The direct access type on which the load file resides

**Loader Utility Quiz** Identify the required job control to invoke the loader utility; select the operands, and code the utility control card to load an emulation program per the following:

*OS-OS/VS*

The 3704 is to be loaded at subchannel address 017 with a DD card labeled CC1.

*DOS-DOS/VS*

SYS007 defines the controller to be loaded at 017.
SYS008 defines the diagnostic routine.
SYS005 defines the emulation program named CC1 to be loaded.
The emulation program is on a DASD 2314.

After you have finished the problem, refer to the solution given in Appendix A.

Criterion

If you make more than one error on the control card or operands, you should review the reading assignment.

## Dump Utility

### Objective

Upon completion of this topic, the student should be able to select and code the utility control card and operands to dump the emulation program.

**Dump Utility Introduction** This topic provides the method of obtaining a dump of the emulation program for analysis, if some specification or other type of error occurs. If the storage address is known, you can also use a dump to locate control blocks and modify options from the communications controller panel.

At the completion of the reading assignment, you will be asked to identify the required job control and code the dump utility card to dump an emulation program.

The dump utility program is one of two methods of obtaining the storage contents of a 3704 or 3705 communications controller. The dynamic dump option (DYNADMP operand of the BUILD macro) allows the contents of storage to be dumped without terminating the emulation program. This option which requires more code and storage than the dump utility, overlays fixed program code for dumping storage.

The independent dump utility program is used to dump the storage contents of a 3704 or 3705 communications controller. The utility accomplishes the dump in two steps:

1. The storage contents of the controller are copied to a direct access data set (SYSUT2).
2. A printable copy of controller storage is produced and placed on a sequential output data set (SYSPRINT). The SYSUT2 data set from step 1 serves as input to step 2.

The dump program is invoked by job control and the two steps of the utility appear to be one job step; however, the second step may be invoked independently by job control to provide multiple printed copies.

The dump program consists of multiple load modules. Most of these modules are executed in the host processor; one module is executed in the communications controller.

Dumping from the controller to the direct access data set is performed by step 1 of the dump utility. This step first transfers into the communications controller a module containing the utility code needed for the controller to participate in the dumping process. (This module is contained within the dump program in the host processor until transferred to the controller via an initial program load (IPL) command).

Step 1 always transfers the entire contents of controller storage and local storage registers to the host processor, which places them on a direct access data set. However, a small portion of the data (in addresses hexadecimal 0 to 79F) is overlaid by the dump utility. The storage area does appear in the listing but consists of read-only storage (ROS) or unmodified IBM-provided dump utility code. The contents of the controller's external registers are not transferred to the host processor. If the contents of these registers must be examined, the registers must be displayed on the operator panel of the controller and the contents noted before the dump utility is invoked.

When step 1 has been completed, the program informs the host operator. At this point the controller is idle and can be reloaded with the loader utility.

The second step of the dump utility converts all or a selected part of the dumped data to printable form, then places the data on a sequential output data set. The output listing shows the hexadecimal representation of the contents of controller storage and the registers and gives the character equivalents of all EBCDIC bit patterns that represent characters. Two options are available for the dump output: mnemonic operation codes may be shown or omitted, and either the complete contents of storage may be listed or any specified portion of storage may be selected for printing.

**Dump Utility Control Card** The dump program requires one control statement, DUMP. This statement specifies the range of storage addresses to be printed and can request the optional printing of mnemonic operation codes.

The operands of the control statement are:

FROMADDR=address-- specifies the lower limit of the controller storage to appear on the listing. If you omit FROMADDR, the listing starts at address X'200'.

TOADDR=address-- specifies the upper limit of the controller storage to appear on the listing. If you omit TOADDR, the listing ends at the upper limit of storage.

MNEMONIC=Y | N-- (default of N) selects the optional printing of mnemonic operation codes (Y) or no printing of mnemonics (N). If mnemonic codes are selected, each normal line of print results in three print lines, a space, mnemonic code, and the contents of storage.

The job control requirements of the dump utility are operating system-dependent. Select the appropriate following reading assignment for your interests and background.

**Dump Utility for OS/VS** To use the dump utility to both dump and print the contents of controller storage, you provide the job control statements for the first step and the DUMP control statement for the second step. The first step generates the required control statements for the second step. The statements are illustrated in Figure 3.5.

```
//jobname    JOB
//stepname   EXEC   PGM=IFLREAD
//SYSUT1     DD     UNIT=address of the controller to be
                    dumped.
//SYSUT2     DD     UNIT=address of DASD work data set.
//SYSPRINT   DD     Sequential data set for dump listing.
//SYSIN      DD     Source of utility control statement.
```

**Figure 3.5 Dump utility job control for OS-OS/VS.**

The dump utility card without mnemonics and printing addresses from X'200' to high storage requires no operands. A control card with operands to select addresses of X'680' to high storage with mnemonics is coded as follows:

```
DUMP FROMADDR=680,MNEMONIC=Y
```

If the DASD work data set is available, the disk file can be selectively printed. All of the contents of storage were written to disk by IFLREAD; any portion of the disk data set can be selected for printing multiple times.

To reprint storage data (disk to print only), the //SYSUT1 DD card may be deleted and the following change made to the previous job control:

```
//stepname EXEC PGM=IFLDUMP
```

**Dump Utility for DOS/VS** Note: The DOS (nonVS) utility does not use a direct access device but formats and prints the contents of controller storage as the data is read from controller storage. The DOS utility is IFUDUMP to invoke the dump. Refer to *IBM 3704 and 3705 Emulation Program Generation and Utilities Guide and Reference Manual* (GC30-3002), Chapter 7: Emulation Program Utilities under DOS.

To use the dump utility to both dump and print the contents of controller storage, you must provide the job control statements for the first step and the DUMP statement for the second step. The first step generates the required control statements for the second step. The job control statements are illustrated in Figure 3.6

```
// JOB
// ASSGN    SYS007    Identifies the controller to be dumped.
                      SYS007 is required.
// DLBL                Identifies the DASD work file to
// EXTENT   SYS008     contain the dumped storage. SYS008
// ASSGN    SYS008     is required.
// ASSGN    SYSLST     The sequential output of controller
                      storage.
// EXEC     IFUREAD    The program reads from SYS007 and
                      writes to SYS008; reads from SYS008
                      and writes to SYSLST.
```

**Figure 3.6 Dump utility job control for DOS-DOS/VS.**

The DUMP control card does not require any operands to print storage from X'200' to high storage without mnemonics. An example of a dump utility card to print storage from X'760' to high storage with mnemonics is as follows:

```
DUMP FROMADDR=760,MNEMONIC=Y
```

If the DASD work file is available, all of storage is written to disk and can be selectively printed. The changes to the job control are as follows:

```
// EXEC IFUDUMP  Selects step two,
                 reading from SYS008
                 and writing to SYSLST.
```

The SYS007 assignment is not used.

**Dump Utility Review** The dump utility is a two-step operation (except for non-VS DOS). Step 1 reads all of controller storage and writes to a DASD data set. The general registers are included, but if external registers are desired it is necessary to display them from the panel prior to invoking the dump utility. Step 2, which can be invoked separately, but is automatically initiated at the end of step 1, reads selected storage addresses and formats the output to a sequential data set.

**Dump Utility Quiz** Identify the job control cards and code the utility control card to invoke the dump utility. Dump a 16K (3FFF hexadecimal) controller at sub-channel 017, without mnemonics. The low address dumped should be hexadecimal 680.

After you have finished coding the problem, refer to the solution in Appendix A.

Criterion

If you fail to identify properly the required DD cards (OS/VS) or SYS assignments, or if you have any errors on the dump utility control card, you should review this section.

## Dynamic Dump Utility

### Objective

Upon completion of this topic, the student should be able to initiate the dynamic dump utility and print the output of the dynamic dump.

**Dynamic Dump Utility Introduction** The dynamic dump utility is an optional utility which provides certain services useful in debugging. Without terminating the execution of the emulation program, the utility can be used to:

1. Obtain the following:
   - -- a storage dump of the communications controller.
   - -- a display of up to 144 bytes of communications controller storage on the host operator's console.
   - -- a dump of the emulation program line trace table.
2. Activate or deactivate the emulation program line trace function.
3. Obtain a dynamic dump of emulation program trace table entries as the entries are made in the trace table.

The controller module of the dynamic dump program uses the NSC or an emulation subchannel within the range recognized by the emulation program and not used by a line or the IPL address. (The IPL address is not allowed in a combination emulation and network control program.) A DASD data set must be allocated for the maximum data received when a trace is active. The maximum is 512 byte records for all of storage, plus one record for registers. If the line trace is used, the DASD space is required for the quantity of data to be traced.

Control statements are used to request the various functions of the dynamic dump utility. These control statements may reside in the SYSIN data set (input stream) or may be entered via the operator's console.

Initially, the dynamic dump utility reads control statements from the SYSIN data set until either an END statement or a PAUSE statement is read. The PAUSE statement instructs the dynamic dump utility to read control statements from the operator's console only, until either an END statement or a SYSIN statement is read. The SYSIN statement is the opposite of the PAUSE statement; SYSIN instructs the dynamic dump utility to return to the SYSIN data set for control statements (beginning with the next statement after the last PAUSE statement). An END statement encountered in the SYSIN data set or entered from the operator's console causes the dump utility to terminate.

The dynamic dump utility is used when trouble or error conditions indicate that a dynamic dump of controller storage is desirable to help in isolating and fixing a problem.

The dynamic dump utility consists of two modules. One module resides in the host processor (as load module IFLSVEP for OS/VS and IFUSVEP for DOS/VS); the other resides in the controller as part of the emulation program. The controller module is included in the emulation program only if DYNADMP=YES is specified on the BUILD macro during program generation. The host and controller modules communicate with each other to transfer specified controller storage to the host module. If the DISPLAY command is used to enter a request, the transferred storage is displayed at the operator's console; otherwise, the host module writes the received storage to the work data set in 512-byte blocks. You may then invoke the PRINT facility of the dynamic dump utility to print the contents of this work data set.

When a particular user request has been completed, the host module of the dynamic dump utility issues a message on the operator console.

**Dynamic Dump Utility Commands** The utility control statements can be issued as a full identifier or in an abbreviated form of only the first two letters. If the commands are provided in the input stream, they may optionally have a one- to eight-character label, beginning with an alphabetic character. Commands from the console may not have a label. The full identifier and explanation of the commands are as follows:

*DYNADMP* The DYNADMP statement requests a dump of the entire controller storage or of a specified portion. The controller does not become idle or require reloading. The valid operands are:

DYNAMIC. Specifies that the trace table is to be dumped dynamically as entries are made. This type of dump requires operator intervention to stop the trace. A trace must be started on a communication line via the control panel of the controller or via the dynamic dump facility (the OPTION control statement) before a dynamic trace can be started.

STORAGE. Specifies that the entire contents of controller storage are to be dumped. The execution of the emulation program

continues both during and after the dump operation.

TABLE. Specifies that only the trace table portion of the controller storage is to be dumped.

If no operand is specified, a full storage dump is produced.

*DISPLAY*

The DISPLAY statement is used to request a display of a portion of controller storage on the operator console at the host processor. The valid operands are:

hhhhh,n-- hhhhh specifies the beginning address (in hexadecimal) of storage displayed. The n value specifies the number of lines (16 bytes per line) to be displayed. The n value defaults to a value of 1 and has a valid range of 1 to 9.

*PRINT*

The PRINT statement requests that a printout (32 bytes of storage per line) of the entire work data set be sent to the print data set. (SYSPRINT for OS/VS or SYSLST for DOS/VS). The only operand is as follows:

START=hh:mm:ss-- Specifies that only those trace entries which were written to the work data set after time hh:mm:ss (hours:minutes:seconds) are to be printed. If the START operand is omitted, the entire work data set is printed. If you specify a START operand and there are storage dumps in the work data set with the trace blocks, then these storage dumps are also printed. Storage dumps are not time stamped.

*OPTION*

The OPTION statement starts, stops, or alters the program interrupt levels being traced. Level 2 interrupts (line data) or level 3 interrupts (timeout complete or channel data, such as initial selection, data, and status) or both can be traced. Level 1 error log entries are traced continuously after a level 3 trace is started. The operand is ABCDE with the values illustrated in Figure 3.7.

| A FUNCTION | BC DATA | DE BYTES | MEANING |
|---|---|---|---|
| 2 | xx | yy | Start L2 trace on subchannel range xx to yy |
| 3 | xx | yy | Stop L2 trace on subchannel range xx to yy |
| 4 | 10 | xx | Start L2 trace on subchannel xx |
| 4 | 11 | xx | Stop L2 trace on subchannel xx |
| 4 | 20 | xx | Start L3 trace on subchannel xx |
| 4 | 21 | xx | Stop L3 trace on subchannel xx |
| 4 | 30 | xx | Start L2 and L3 trace on subchannel xx |
| 4 | 31 | xx | Stop L2 and L3 trace on subchannel xx |
| 4 | 70 | 00 | Start L3 trace on trace defined subchannels |
| 4 | 71 | 00 | Stop L3 trace on trace defined subchannels |
| 4 | 70 | FF | Start L3 trace on all subchannels |
| 4 | 71 | FF | Stop L3 trace on all subchannels |

**Figure 3.7 Dynamic dump OPTION statement values.**

*PAUSE*

The PAUSE statement allows control statements to be entered at the console of the host processor after the PAUSE statement is read from the input job stream or entered from the console.

There are no operands on the PAUSE statement. A PAUSE is ended by an operator-entered command of END to terminate the utility or by a command of SYSIN to return command input to the input job stream.

*END*

The END statement specifies the end of job and causes termination of the dynamic dump program, but only after printing current trace data which has been received.

*SYSIN*

The SYSIN statement is used by an operator to cause control statements to be read from the input stream. A job stream command of PAUSE must precede any attempt to enter input from an operator console.

The information which follows is operating system-dependent. Select the appropriate reading according to your interest and background.

**Dynamic Dump for OS/VS** The job control requirements for initiating the dynamic dump are illustrated in Figure 3.8.

| | | |
|---|---|---|
| //jobname | JOB | |
| //stepname | EXEC | PGM=IFLSVEP, PARM='LINECOUNT=nn' nn default 55. |
| //SYSPRINT | DD | Sequential output file. |
| //SYSUT1 | DD | Defines the subchannel to the controller for dynadump. |
| //SYSUT2 | DD | Work data set (DASD) for dumped data. |
| //SYSIN | DD | Control statements (If control statements are to be provided from the console, a PAUSE statement must be provided in SYSIN.) |

**Figure 3.8 Dynamic dump utility job control for OS-OS/VS.**

**Dynamic Dump for DOS/VS** The job control required to invoke the dynamic dump utility is illustrated in Figure 3.9.

| | | |
|---|---|---|
| // JOB | | |
| // ASSGN | SYSLST | Defines the output file. |
| // ASSGN | SYS011 | Defines the dymanic dump subchannel. |
| // ASSGN | SYS010 | Defines the DASD work file. |
| // ASSGN | SYSIPT | Defines the control statement file (If control statements are to be entered from the console, a PAUSE statement is required from SYSIPT). |
| // EXEC | IFUSVEP | Initiates the dynamic dump. |

**Figure 3.9 Dynamic dump job control for DOS-DOS/VS.**

The following Logical Input/Output Control System (LIOCS) modules must be cataloged in the relocatable library:

```
IJCFZIW0
IJDFAZZW
IJFUZZWZ
```

If necessary, the following macros can be assembled to provide the above modules:

```
CDMOD TYPEFLE=INPUT,
      WORKA=YES,
      SEPASMB=YES

PRMOD CTLCHR=ASA,
      WORKA=YES,
      SEPASMB=YES

MTMOD RECFORM=UNDEF,
      WORKA=YES,
      SEPASMB=YES
```

The controller physical unit block must indicate a TP device for SVC 27 (HALT I/O) to work. A TP device can be specified by coding 2701 on the ADD command.

The supervisor must be generated with AP=YES for the POST instruction to work, and TP=BTAM or TP=QTAM must be specified for the HALT I/O (SVC 27) instruction to be supported.

**Dynamic Dump Utility Review** The dynamic dump provides for storage dumps, storage displays of up to 144 characters, and line trace printout during trace execution. The trace is invoked by job control, but control commands may be read from the job stream and/or the operator console. The trace is concurrent with execution of the emulation program. The emulation program continues operation during and following execution of the dynamic dump without the emulation program having to be reloaded.

**Dynamic Dump Utility Quiz** Identify the required data files which must be defined by job control and code the utility control statements for invoking the dynamic dump.

Code the dynamic dump utility control statements in sequence to perform the following:

1. Start a trace for levels 2 and 3 on subchannel 010.
2. Switch the source of commands to the console.

If the host console returns control to the job stream, provide the following additional commands.

1. Stop the trace for levels 2 and 3 on subchannel 010.
2. Terminate the dynamic dump.

After you have finished coding the problem, refer to the solution in Appendix A.

Criterion

You should be able to identify the control cards required to invoke the dynamic dump and to code the utility control cards for the problem without any errors. If made an error, you should review this section.

# Appendix A:
## Quiz Solutions

### Introduction Quiz

1. True
2. False. Only the dynamic dump facility allows concurrent dumping and continued emulation program execution.
3. True
4. False. Emulation programming may use only the type 1 or type 4 CA.
5. True
6. True
7. 255
8. 4

### Definition Quiz

1. a
2. True
3. True
4. False
5. True

### BUILD Macro Quiz

The solution that follows is by operating system type. Refer to the appropriate section to check your coding of the BUILD macro.

This problem allows you to verify your understanding of the emulation program and generation process. For an accurate self-evaluation on the subject matter, do not check the answers until you have coded all operands of the BUILD macro problem.

**DOS/VS BUILD Macro** The following operands may occur in any sequence:

symbol

An optional label of one to eight characters, the generation procedure does not check the symbol for validity. If coded, the symbol must start in column 1.

CA=TYPE1

On a 3704, only the TYPE1 operand is valid.

DYNADMP=YES

This facility was to be included. When the dynamic dump facility is wanted (to allow trace entries or storage to be sent to the host without stopping the emulation program), this operand should be coded YES.

HICHAN=17

The subchannel range of eight starting at 10 results in a high channel address of 17.

JOBCARD=

(OS/VS or DOS/VS NCP only)

LESIZE=

(OS/VS only)

LINETRC=(YES,n)

As the default is YES, this operand is optional. All optional trace facilities were to be included. The n value is the maximum number of lines to be traced concurrently (all lines is the default).

LOADLIB=

(OS/VS only)

LOCHAN=10

The lowest subchannel address was specified as 10.

MODEL=3704

A 3704 was defined in the problem, so coding 3705 is an error.

NEWNAME=

(OS/VS or DOS/VS NCP only)

OBJLIB=

(OS/VS only)

OPCSB2=NO

The four-byte buffers are all that is required by the problem definition. The default is NO.

QUALIFY=

(OS/VS only)

TEST=YES

This operand is required by the problem definition requesting all optional trace and test facilities. The default of NO or coded TEST=NO is not correct for this problem. TEST=YES operand includes the panel-initiated line test function.

TYPGEN=EP

This operand is optional as the default generates an emulation program. This operand is not valid for a nonVS system.

TYPSYS=DOS

This operand defaults to generating an OS or OS/VS program. This operand is required for a DOS or DOS/VS system.

UNIT=

(OS/VS only)

UT1=, UT2=, UT3=

(OS/VS only)

**OS/VS BUILD Macro** The following operands may occur in any sequence. If you did not code the operand correctly, please reread the appropriate section in the reading material.

symbol An optional label of one to eight characters; the generation process does not check the symbol for validity. If coded, the symbol must start in column 1.

CA=TYPE1

TYPE1 is the only valid operand for the 3704.

DYNADMP=YES

This facility was to be included. When the dynamic dump facility is wanted (to allow trace entries or storage to be sent to the host without stopping the emulation program), this operand should be coded YES.

HICHAN=17

The subchannel range of eight starting at address 10 results in a high channel address of 17.

JOBCARD=

This parameter was not specified in the problem. If you coded NO, you must provide a JOB card prior to the stage 2 generation. If you coded YES, a job card will be created for stage 2. These are the only valid operands for an emulation program.

LESIZE=

This parameter was not specified in the problem If this operand is not coded, no REGION operand is generated for the stage 2 linkage editor job steps. If this operand is coded, it must be greater than 200 and less than 16384 (16,384K bytes). This operand is required only for an operating system with regions.

LINETRC=(YES,n)

The problem definition requested that all optional trace and test facilities be included. The YES operand is the default, so you may either omit the operand or code YES. LINETRC=NO is incorrect. The operand n defines the number of lines to be traced concurrently (default of all defined lines).

LOADLIB=LOAD3704

The load library data set was specified as SYS1.LOAD3704 and the SYS1 is provided on the QUALIFY operand.

LOCHAN=10

The lowest subchannel address was specified as 10.

MODEL=3704

A 3704 was defined in the problem, so coding 3705 is an error.

NEWNAME=TEST1

The member name of the program was specified as TEST1, which is placed in the SYS1.LOAD3704 library. This operand is required because the default name of EP001 is incorrect for the problem specification.

OBJLIB=OBJ3704

The object library was to be SYS1.OBJ3704 and the SYS1 is specified on the QUALIFY operand.

OPCSB2=NO

The four-byte buffers are all that is required by the problem definition. NO is the default.

QUALIFY=SYS1

SYS1 is the default value. You could either code SYS1 or omit this operand for a correct answer. This qualifier will be used for all the data sets specified on the BUILD macro.

TEST=YES

This operand is required by the problem definition requesting all optional trace and test facilities. The default value of NO or coding NO is incorrect. This operand includes the panel-initiated line test function.

TYPGEN=EP

If this operand is omitted, the default generates an emulation program. This operand is not valid on a nonVS system.

TYPSYS=OS

This operand can be omitted, as OS is the default.

UNIT=

This operand was not specified in the problem. The default or any valid unit specification for direct access devices is valid.

UT1=, UT2=, UT3=

The specification of any of these operands requires you to use preallocated, cataloged data sets for assembler and linkage editor work files. If you omitted these operands, temporary data sets are created for assembler and linkage editor work files. The problem did not specify which method to use, but if you coded any of these operands you must allocate and catalog the data sets prior to stage 2 generation.

## CSB Macro Quiz

The CSB macro is the same for DOS/VS and OS/VS. Only one CSB macro is required for the defined configuration.

symbol

An optional label of one to eight characters. The generation process does not check the symbol for validity. If coded, the symbol must start in column 1.

SPEED=134

The internal clocking is specified as a single clock of 134.5 bps. This clock value is stated in the operand merely as 134, 134 without the tenths value. The 134 may be coded as (134).

WRAPLN=020

The wrap line for diagnostics was defined to be the first scanner address, which for a type 2 scanner is 020.

MOD=0

This operand defaults to zero and may be omitted for this problem. A zero value is the only valid entry for a 3704. If more than one CSB macro is required for a 3705 type 2 or 3 communications scanner, this operand specifies which scanner is defined by each of the macros.

TYPE=TYPE2

This operand is required, as a type 1 communications scanner is assumed for a 3704.

## GROUP Macro Quiz

Two GROUP macros are required for the emulation problem: one for the two lines of BSC and one for the two start-stop lines. The GROUP macros must follow the BUILD and CSB macros; however, there is no sequence requirement for the GROUP macros themselves. In other words, GROUP macros may occur in any sequence in an emulation program.

It is not wrong if you coded a GROUP macro for each line; however, each GROUP definition generates a control block requiring storage space. Thus from a storage requirement, it is best to code only two GROUP macros.

Please make sure you coded the GROUP macros on separate coding forms. You must have coded the LINE macros following the appropriate GROUP macro and some LINE macro operands may be promoted to the GROUP level.

## BSC GROUP

symbol

The symbol provides a one to eight character name for the line group. This label is required and may be any valid assembler language symbol, except that the first character may not be $. The problem did not specify a label to be used; any assembler language symbol is valid except those starting with $ or those used as labels elsewhere in this program.

CHAREC=

This operand is not used for BSC lines and should have been omitted. CHAREC is coded only for TWX terminals.

DELAY=NO

This operand is not used for SBC lines and should have been omitted. DELAY is coded YES only for world trade terminals requiring a long line turnaround.

DIAL=NO

This operand is optional, because the default of NO is correct for the nonswitched BSC lines specified in the problem.

EOB=

This operand is not used for BSC lines and should have been omitted. EOB applies to World Trade teletypewriters only.

EOT=

This operand is not used for BSC lines and should have been omitted. EOT applies to World Trade teletypewriters only.

LNCTL=BSC

This operand is required for BSC lines. The default specifies a start-stop group.

REPLYTO=

This operand is only valid for start-stop lines and should have been omitted.

TEXTTO=

This operand is only valid for start-stop lines and should have been omitted.

## Start-Stop GROUP

symbol

The symbol provides a one to eight character label for the line group. This label is required and may be any valid assembler language symbol, except that the first character may not be $. The problem did not specify a label to be used; any assembler language symbol is valid except those starting with $ or those used as labels elsewhere in this program.

CHAREC=

This operand defines the text-ending characters for teletypewriter terminals. CHAREC is not used for 2741 terminals and should have been omitted.

DELAY=NO

This operand is used only for world trade teletypewriters requiring a long line turnaround. This operand should have been omitted.

DIAL=YES

This operand is required for switched lines. The default of NO is an error by the problem definition.

EOB=

This operand specifies the character sequence used to indicate end-of-block for World Trade teletypewriters. EOB is not used for 2741 terminals.

EOT=

This operand specifies the character sequence used to indicate end-of-transmission for World Trade teletypewriters. EOT is not used for 2741 terminals.

LNCTL=SS

The SS default selects start-stop mode of line control. You can omit this operand or code SS for the start-stop group.

REPLYTO=3.0

The default (3.0) was specified for all timeout values in the problem definition. Either the default or coded operand of 3.0 is correct.

TEXTTO=25.6

The default (25.6) was specified for all timeout values in the problem definition. Either the default or coded operand of 25.6 is correct.

## LINE Macro Quiz

The LINE macros must be coded following the appropriate GROUP macro. BSC LINE definitions must follow the BSC GROUP definition; the start-stop LINE definitions must follow the start-stop GROUP definition. There is no required sequence of specific groups within an emulation program except that all GROUP macro must follow the last CSB macro and precede the GENEND macro.

**BSC LINE Macros** NOTE: The following operands are promotable and may be coded on either the GROUP or LINE macros. The operands required on the LINE macro follow the last promotable operand.

BUFSIZE=

This operand applies only to type 3 communications scanners and should have been omitted.

CHECK=

This operand applies only to switched start-stop lines and should have been omitted.

CHNPRI=NORMAL

This operand should have been omitted or coded with the default. The HIGH operand should be coded only for lines with a data rate of 19,200 bps or greater.

CLOCKNG=EXT

Modem clocking was defined in the problem as external. This operand defaults to EXT when the LINE macro follows a BSC GROUP macro. Check the spelling of your coding to be sure you have omitted the I in CLOCKNG.

CODE=EBCDIC

The default or a coded operand of EBCDIC is correct.

CU=2703

The default or a coded operand of 2703 is correct.

CUTYPE=3271

The default or a coded operand of 3271 is correct.

DATRATE=LOW

The problem specified that all modems are single speed, so the default of LOW or a coded value of LOW should be specified.

DISABLE=NO

The problem specified that a 'long disable timeout' was not required, so either the default or a coded NO operand should be used.

DUPLEX=FULL

The problem stated that all lines are half-duplex operation with a full-duplex interface connection between the line set and the modem. All emulation lines work

as half-duplex, but the interface at the line set may be either half- or full-duplex. Since the default is HALF, this operand must be coded FULL.

FEATURE=(NODUALCD)

The terminals are specified as EBCDIC only. SINCE NODUALCD is the default, the operand can be omitted. NODUALCD is the only suboperand which is valid for BSC lines.

INTPRI=1

The problem requested a line priority of 1 for BSC lines and 1 is the default value. The operand can be coded or omitted.

MODEM=OPTION2

The problem definition states that the modems are disabled until enabled by command from the access method. The default value or coded value of OPTION2 is correct.

MULTI=

This operand should have been omitted, as it applies only to the IBM 2845 or 2848.

NEWSYNC=YES

The problem specified the modem as having the 'newsync' signal. The operand default is YES for a group with LNCTL=BSC, DUPLEX=FULL, CLOCKNG=EXT, and DIAL=NO, so this operand can be omitted.

PAD=YES

This operand can be omitted; the default is YES. The problem definition specified that the first four bits of trailing pad characters are to be checked for 1's.

QUIET=NO

A 'long disable timeout' is not required by the problem, so either the default or a coded value of NO should be used.

RING=

The operand does not apply to a nonswitched line and should have been omitted.

SPEED=2400

This operand is required as the specification of a 2400 bps BSC line was asked for in the problem.

TADDR=NONE

The operand does not apply unless this controller is a tributary station on a BSC line. This operand should have been omitted or coded NONE.

TERM=

This operand may be omitted. When CUTYPE operand is coded the TERM operand is ignored.

UNITXC=

This operand does not apply to BSC lines.

Nonpromotable operands which must be coded on the LINE macros following the BSC GROUP macro are:

symbol

Provides a one to eight character label for the line defined by this LINE macro. This label is required and may be any valid assembler language symbol, except that the first character may not be $. The problem did not specify a label to be used. Any symbol is valid except those starting with $ or those used as labels elsewhere in this program. Two LINE macros are required following the BSC GROUP macro, each with a label.

ADDRESS=(020,10)
ADDRESS=(021,11)

One of these operands should be coded on either of the two LINE macros and the alternate on the other LINE macro. These entries connect line scanner address 020 to subchannel address 10 and line scanner address 021 to subchannel 11. It is also valid to have (020,11) and (021,10) as operands.

AUTO=

This operand is not valid for a nonswitched line or lines used for callin only. AUTO should have been omitted.

DUALCOM=

This operand is not valid except for dual communication interface functions of a 2701. DUALCOM should be omitted or coded NONE.

**Start/Stop LINE Macros** The following are the promotable operands which can be coded on either the GROUP OR LINE macros. The operands which must be coded on the LINE macro immediately follow the last promotable operand.

BUFSIZE=

This operand applies only to type 3 communications scanners and should have been omitted.

CHECK=

Data carrier detect monitoring was not required, so this operand should have been omitted or coded with the default of NODCD.

CHNPRI=NORMAL

This operand should have been omitted or coded with the default of NORMAL. The HIGH operand applies only to lines with a data rate of 19,200 bps or greater.

CLOCKNG=INT

Internal clocking was defined in the problem. This operand defaults to INT when it follows a start-stop

group, so this operand can be coded or omitted. Check the spelling of your coding to be sure you omitted the I in CLOCKNG.

CODE=

This operand applies only to BSC lines and should have been omitted.

CU=2703

The default is 2703, so either an omitted operand or coded value of 2703 is correct.

CUTYPE=

This operand applies only to cluster-type stations. Since the 2741 is not a cluster type, this operand should have been omitted.

DATRATE=LOW

The problem specified that all modems are single speed, so either the default of LOW or coded value of LOW is correct.

DISABLE=NO

The problem specified that a 'long disable timeout' was not required, so either the default or a coded NO operand should be used.

DUPLEX=FULL

The problem stated that all lines are half-duplex operation with a full-duplex interface connection between the line set and modem. All emulation lines work as half-duplex, but the interface at the line set may be either half- or full-duplex. Since the default is HALF, this operand must be coded FULL.

FEATURE=(NOIMEND)

The suboperands of the FEATURE operand either do not apply or the default is appropriate, so this operand can be omitted. The DUALCODE applies only to BSC. The NOIMEND operand is the default, indicating that the immediate end feature is not required. If paper tape transmission were being received with imbedded EOT's, then IMEND would be required to recognize the EOT and immediate continuation of data characters. The LRC or NOLRC for the record-checking feature does not apply to 2741 terminals. SPACE or NOSPACE operand applies to teletypewriter terminals only.

INTPRI=0

The problem requested a line priority of 0 for start/stop lines. The default is 1, so this operand is required.

MODEM=OPTION2

The problem definition states that the modems are disabled until enabled by command from the access method. The default value of OPTION2 is correct, so this operand can be omitted.

MULTI=

This operand applies only ti IBM 2845 or 2848, so it should have been omitted.

NEWSYNC=

The operand is not valid for start/stop lines, and should have been omitted.

PAD=YES

The first four bits of trailing pad characters should be checked for 1 bits. This operand must be coded YES or omitted for the default.

QUIET=NO

A 'long disable timeout' is not required by the problem, so either the default or a coded value of NO should be used.

RING=NO

The operand can be coded NO or the default used to specify that the line set/modem connection does not include the 'ring indicator interface' lead connection. This operand is not used in the United States or Canada.

SPEED=134

The line speed was specified as 134.5 for the start-stop lines. This operand is required.

TADDR=NONE

The operand does not apply unless this controller is a tributary station on a BSC line. This operand should have been omitted or coded NONE.

TERM=2741

The terminal type was specified as 2741 and TERM must be coded.

UNITXC=YES

The EOT from a line should provide the Unit Exception status to the host. The coded YES operand or default value may be used.

The nonpromotable operands which must be coded on the LINE macro are listed below. The LINE macro must immediately follow the GROUP macro for that particular line.

symbol

The symbol provides a one to eight character label for the line defined by this LINE macro. This label is required, and may be any valid assembler language symbol, except that the first character may not be $. The problem did not specify a label, so any assembler

language symbol is valid except those starting with $ or those used elsewhere in this program.

ADDRESS=(022,12)
ADDRESS=(023,13)

One of these operands should be coded on either of the two LINE macros and the alternate on the other LINE macro. These entries connect line scanner address 022 to subchannel address 12 and line scanner address 023 to subchannel address 13. The operands can also be coded (023,12) and (022,13).

AUTO=

The problem defines callin lines only, so this operand should be omitted. AUTO is valid only for switched callout lines with the Automatic Calling Unit (ACU) Interface Address. The operand provides the scanner address of the ACU.

DUALCOM=

This operand is valid only for the dual communication interface function of a 2701 and should have been omitted.

### GENEND Macro Quiz

The GENEND macro must follow the coding of all other problems which were previously coded in this problem set.

symbol

The symbol is an optional label of one to eight characters, first character alphabetic, and may not be defined elsewhere in the program.

SCANCTL=(n,0000)

The problem specified that address substitution and upper scan limit were not required, so this operand can be omitted. An n value of 0 scans 96 addresses; however, any value of 0 to 3 scans the four addresses at an appropriate rate. The mask of 0000 (the default) selects no address substitution.

### Generation Quiz

1. The 3704 and 3705 assembler names are:

   DOS - IFTASM
   DOS/VS - IFZASM
   OS - IFKASM
   OS/VS - CSAX00

2. GC30-3003. Assembler diagnostics are provided in GC30-3003. The diagnostic messages resulting from invalid assembler macro coding are provided in GC30-3008 for OS/VS and DOS/VS, GC30-3002 for OS and DOS.

3. The same data sets as required by the OS/VS or DOS/VS assembler.

4. The output of stage 1 emulation generation assembly is:

DOS/VS: Control blocks and a listing of INCLUDE cards naming the modules to be linked with the control blocks to create the load module.

OS/VS: A job stream consisting of multiple assembly and linkage editor steps which create the load module.

## Loader Utility Quiz

### OS-OS/VS

```
//LOAD      JOB
//          EXEC  PGM=IFLOADRN
//SYSPRINT  DD    SYSOUT=A
//SYSUT1    DD    DSN=SYS1.LOAD3705,
                  DISP=SHR
//SYSUT3    DD    DSN=SYS1.LINKLIB,
                  DISP=SHR
//CC1       DD    UNIT=017
//SYSIN     DD    *
 LOAD  LOADMOD=TEST1,3705=CC1,DIAG=Y6
```

The following is an explanation of the solution:

The JOB card is standard. The EXEC card names the load module utility of IFLOADRN. SYSOUT names the standard SYSOUT print data set or other sequential output data set.

The library for the output of the generation was specified on the BUILD macro operand of LOADLIB and QUALIFY as SYS1.LOAD3705. The data set had to be cataloged for the generation process.

SYSUT3 names the library of the diagnostic routine, SYS1.LINKLIB.

The CC1 DD label may be any valid label not used on other DD cards of this job step, but the DD label must be the same as the load utility card operand which identifies the DD card of the unit to be loaded.

The SYSIN DD card identifies the data set to obtain the load utility card.

The load utility card LOAD field must not start in column 1. LOADMOD=TEST1 is required to identify the emulation program name you coded on the BUILD macro operand of NEWNAME=TEST1. The label of CC1 can be any valid label which matches the name of the DD card defining the IPL address. The //SYSUT3 card identified the diagnostic routine library. DIAG=Y6 provides diagnostics for machine addressing for 64K storage or less with the 16-bit addressing. (DIAG=NO bypasses the diagnostic routine and loads your program immediately.

### DOS-DOS/VS

```
//        JOB

//        ASSGN  SYS007,X'017'
//        DLBL   DIAGFLE, --
//        EXTENT SYS008,  --
//        ASSGN  SYS008,  --

//        DLBL   TEST1,   --
//        EXTENT SYS005,  --
//        ASSGN  SYS005,  --

//        EXEC   IFULOAD

   LOAD   LOADMOD=TEST1,
          3705=SYS007,
          DIAG=Y6,
          DEVICE=2314
```

Note: The LOAD card must be contained on one card or 80-character record.

The following is an explanation of the solution:

The JOB card is standard.

The subchannel address for IPL is 017. SYS007 is the LUB used by the load utility because of the operand on the LOAD utility control card.

DIAGFLE identifies the location of the diagnostic routine to be used at load time. The diagnostic routine is provided with the load utility and must be placed in a sequential file by the user.

TEST1 may be any valid label, but it must be the same label as the LOAD utility card operand which provides the file name in the operand LOADMOD=.

The load utility is executed by the EXEC IFULOAD.

The LOAD utility statement CC1 operand must match the label on the DLBL first operand which identifies the sequential file containing the generated emulation program. SYS007 can be any logical unit to be used for loading. The Y6 operand specifies the diagnostics are for a machine with addressing of 64K storage or less with 16-bit registers. The DEVICE operand specifies the type of device where the emulation program is to be obtained for loading.

## Dump Utility Quiz

### OS-OS/VS

```
//DUMP   JOB
//       EXEC  PGM=IFLREAD
//SYSUT1 DD    UNIT=017
//SYSUT2 DD    --
//SYSPRINT DD
//SYSIN  DD    *

 DUMP FROMADDR=680,
      TOADDR=3FFF,
      MNEMONIC=N
```

Note: The DUMP utility card must be contained within one card or one 80-character record.

The following is an explanation of the solution:

The JOB card is standard.

The EXEC card identifies the program name, IFLREAD.

SYSUT1 identifies the 017 as the unit address of the controller to be dumped.

SYSUT2 identifies a DASD work data set used as a temporary holding area prior to formatting controller storage for printing.

SYSPRINT specifies the dump listing output device or other sequential output data set.

SYSIN specifies the source of the dump utility control card.

The DUMP utility control card can be coded without any operands except FROMADDR (other operands are defaults).

### DOS-DOS/VS

```
//       JOB   DUMP
//       ASSGN SYS007,X'017'
//       EXEC  IFUDUMP
         DUMP  FROMADDR=680,
               TOADDR=3FFF,
               MNEMONIC=N
```

Note: The DUMP utility card must be contained within one card or one 80-character record.

The following is an explanation of the solution:

The JOB card is standard.

SYS007 must be assigned to the subchannel address of the controller to be dumped.

The EXEC card identifies the dump utility, IFUDUMP.

The DUMP utility control card can be coded without any operands except FROMADDR=680 (other operands are defaults).

## Dynamic Dump Quiz

The job control solutions are illustrated in Figure 3.8 (OS-OS/VS) or Figure 3.9 (DOS-DOS/VS).

The command sequence in the job stream is as follows:

```
OPTION 43010
PAUSE
OPTION 43110
END
```

# Appendix B:

## Types of Stations Supported by the IBM 3704 and 3705 Communications Controllers in Emulation Mode

When attached a host processor channel, the IBM 3704 and 3705 Communications Controllers can communicate in emulation mode with stations of the types listed below. Consult your IBM representative for the specific requirements for support of each of these stations.

### Start-Stop Terminals:

IBM 1030 Data Collection System

IBM 1050 Data Communication System

IBM 1060 Data Communication System

IBM 2260 Display Station (via IBM 2848 Display Control)

IBM 2265 Display Station (via IBM 2848 Display Control)

IBM 2740 Communications Terminal, Models 1 and 2

IBM 2741 Communications Terminal

IBM 2760 Optical Image Unit (via the IBM 2740 Communications Terminal, Model 1)

### Binary Synchronous Terminals:

IBM 2770 Data Communications System

IBM 2780 Data Transmission Terminal (no support for Transcode)

IBM 2970 Models 5 and 8 Banking Terminals (not available in the United States and Canada)

IBM 2972 General Banking Terminal System, Models 8 and 11

IBM 3270 Information Display System

IBM 3650 Retail Store System

IBM 3670 Brokerage Communication System (supported only in the United States and Canada)

IBM 3735 Programmable Buffered Terminal

IBM 3740 Data Entry Terminal

IBM 3780 Data Communication Terminal

### Transmission Control Units:

IBM 2701 Data Adapter Unit (with Synchronous Data Adapter Type II) (notes 1 and 2)

IBM 2703 Transmission Control (with Synchronous Terminal Control) (notes 1 and 2)

IBM 2715 Transmission Control, Model 2 (note 1)

IBM 3704 Communications Controller (note 3)

IBM 3705 Communications Controller (note 3)

---

Note 1 -BSC support only

Note 2 -Supported only when attached locally to an IBM System/360 or System/370

Note 3 -A local controller can communicate in emulation mode with other channel-attached 3704 or 3705 controllers via a binary synchronous communication line.

### Computers (BSC support only except for System/7):

IBM System/3

IBM System/7 (supported as an IBM 2740 Communications Terminal, Model 1, with the Record Check feature; also supported as a BSC station)

IBM System/360, Model 25 (with Integrated Communications Attachment with Synchronous Data Adapter II)

IBM System/370, Model 125 (with Integrated Communications Attachment with Synchronous Data Adapter II)

IBM System/370, Model 135 (with Integrated Communications Attachment with Synchronous Data Adapter II)

IBM 1130 Computing System (with Synchronous Communications Adapter)

IBM 1800 Data Acquisition and Control System (via IBM 1826 Data Adapter Unit with Communications Adapter

IBM 3750 Switching System (not available in the United States and Canada)

The controllers also communicate in emulation mode with the following:

World Trade teleprinters that use CCITT No. 2 or No. 5 code on leased point-to-point, leased multipoint, or switched network communication lines.

Terminals using the following line control disciplines: AT&T 83B3 or WU 115A start-stop code, over point-to-point or multipoint leased telegraph lines; CPT-TWX (33/35) start-stop code over the switched communications network.

Attachment of nonIBM equipment is under provisions of the IBM Multiple Support System Policy.

# Glossary

## List of Abbreviations

| | |
|---|---|
| ASCII | American Standard Code Information Interchange |
| AT & T | American Telephone and Telegraph |
| BCD | Binary Coded Decimal |
| BSC | Binary Synchronous |
| CA | Channel Adapter |
| CPU | Central Processing Unit |
| CS | Communication Scanner |
| EBCDIC | Extended Binary Coded Decimal Interchange Code |
| EP | Emulation Program |
| ESC | Emulation Subchannel (Address) |
| FIGS | Figures (character to shift to numeric on TWX) |
| ID | Identification |
| IPL | Initial Program Load |
| LTRS | Letters (character to shift to alphabetic on TWX) |
| MSLA | Multi-subchannel Line Access |
| NCP | Network Control Program |
| NSC | Native Subchannel (Address) |
| PEP | Partitioned Emulation Program |
| RPQ | Request for Price Quotation |
| TWX | Teletypewriter Exchange |
| WRU | Who-are-you (ID exchange on TWX) |

**Access Method:** A data management technique for transferring data between main storage and input/output units.

**Addressing:** The means whereby the originator or control unit selects the telecommunication device to which a message is going to be sent.

**Attachment Base:** An attachment base is a required feature for support of the 3705 adapters. The type 1 attachment base provides common controls to the central control unit for both the type 1 scanner and the type 1 channel adapter. The type 2 attachment base provides common controls to the central control unit and line addressing controls for the type 2 and type 3 scanners.

**Channel Adapter (CA):** A controller hardware unit which provides attachment of the controller to a System/360 or System/370 Channel.

**Communication Scanner (CS):** A controller hardware unit which provides the interface between line interface bases and the central control unit. The communication scanner monitors the communication lines for service requests.

**Duplex Line:** A communication line having two data paths that allow data to be transmitted over one path while concurrent 'receive' occurs on the second path. Also known as full-duplex.

**Duplex Modem Strapping:** The duplex connection between a line set and a modem. Strapping may be half- or full-duplex for a modem which operates on a line in either half or full-duplex data transmission mode.

**Emulation Program:** A control program for the controllers which allows the user to draw on a library of IBM-supplied modules to emulate the IBM 2701 Data Adapter Unit, the 2702 Transmission Control, and 2703 Transmission Control.

**Full-Duplex Line:** See Duplex Line.

**Full-Duplex Modem Strapping:** See Duplex Modem Strapping.

**Generation Delimiter Macro:** The macro that marks the end of the control program generation input stream (GENEND macro).

**Half-Duplex Line:** A communication line having a single data path over which data can be transmitted in either direction, but not simultaneously. Contrast with duplex line.

**Half-Duplex Modem Strapping:** The half-duplex connection between a line set and a modem. Strapping may be half- or full-duplex for a modem which operates on a line in either half- or full-duplex data transmission mode.

**Host Processor:** The central processing unit to which a controller is attached by a channel and which executes the telecommunications access method to support that controller.

**Initial Test Routine:** A diagnostic program executed in the controller before the control program is loaded. The initial test routine tests the controller hardware for conditions that might cause failure after operation begins.

**Interrupt:** A break in the normal sequence of instruction execution. An interrupt causes an automatic transfer to a preset storage location where appropriate action in taken.

**Line Adapter:** An IBM modem that is a feature of a particular product. Some communications controller line sets include line adapters; others require external modems. See also Modem.

**Line Control Character:** A special character that controls transmission of data over a communication line. For example, line control characters are used to start or end a transmission, to cause transmission error checking to be performed, and to indicate whether a station has data to send or is ready to receive data.

**Line Group:** A group of communication lines by which stations supported by the same line-control discipline are connected to the controller.

**Line Interface Base (LIB):** A controller hardware unit that provides for the attachment of communication lines to the controller.

**Line Set:** A controller hardware unit through which one or more lines are attached to a line interface base.

**Load Module:** A program in a format suitable for loading into storage for execution.

**Longitudinal Redundancy Checking:** Parity checking of the transmission bits to detect transmission errors.

**Network Control Program:** A control program for the controllers, generated by the user from a library of IBM-supplied modules.

**Modem:** (MOdulator-DEModulator) A device that modulates and demodulates signals transmitted over communication facilities. See also Line Adapter.

**Modem Strapping:** See Full-duplex and Half-duplex Modem Strapping.

**Partitioned Emulation Program (PEP):** A control program which provides the functions of both the Emulation Program and Network Control Program. Generated by the user from a library of IBM-supplied modules.

**System Macro:** One of the control program generation macros that provide information pertaining to the entire controller.

**Transmission Code:** A character code for sending information over communication lines.

# Index

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
360 Hamilton Avenue, White Plains, New York 10601
(International)

SR20-4546-2

IBM 3704/5 Communications Controllers Emulation Programming Printed in U.S.A. SR20-4546-2