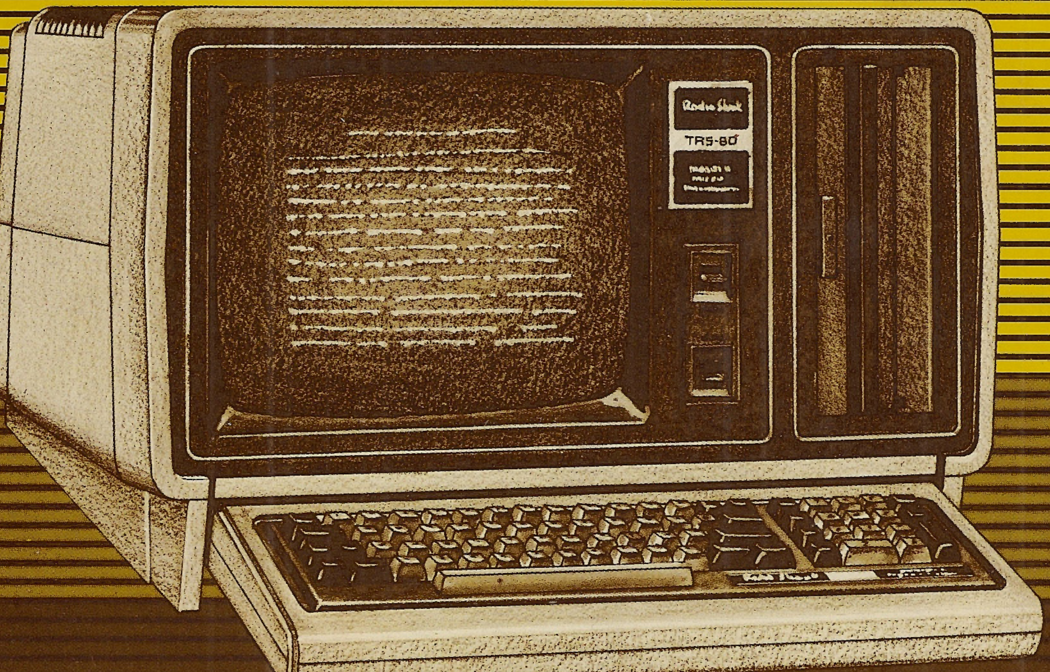
TRS-80® Model II

BiSync Communications
Remote (2770, 2780, 3741, 3780) and CPU Modes



Radio Shack®
The biggest name in little computers™

CUSTOM MANUFACTURED IN THE USA BY RADIO SHACK A DIVISION OF TANDY CORPORATION

TRS-80 Model II BISYNC-80/3780

User's Manual

Catalog Number 26-4716

Reproduction or use, without express written permission from Tandy Corporation, of any portion of this manual is prohibited. While reasonable efforts have been taken in the preparation of this manual to assure its accuracy, Tandy Corporation assumes no liability resulting from any errors or omissions in this manual or from the use of the information obtained herein.

Contents

To Our Customers...

This manual provides the information you need to use BISYNC-80/3780 on the TRS-80 Model II.

If you want more information about the communications protocol used by BISYNC-80/3780, see the following IBM publications:

For details of the Binary Synchronous Communications protocol, refer to the IBM publication:

- GA27-3004 General Information Binary Synchronous Communications

For details of the protocol used by the IBM devices emulated, refer to the IBM publications:

- GA27-3013 System Components Manual: IBM2770 Data Communications System
- GA27-3005 Component Information for the IBM 2780 Data Communication Terminal
- GC21-5071 IBM 3740 BTAM/TCAM Programmer's Guide
- GA27-3065 Component Information for the IBM 3780 Data Communication Terminal

You'll also need to refer to:

- TRS-80 Model II Owner's Manual (Catalog Number 26-4920).
- TRS-80 Model II Editor/Assembler (Catalog Number 26-4702).

IMPORTANT NOTE

It is highly recommended that you make a "safe" copy of the supplied BISYNC-80/3780 diskette before you begin using it. For details on making a copy, see **BACKUP** in your TRS-80 Model II owner's manual.

Notational Conventions

The following conventions are used to show syntax in this manual:

CAPITALS

Any words or characters which are uppercase must be typed in exactly as they appear.

lowercase underline

Fields shown in lowercase underline are variable information that you must substitute a value for.

<KEYBOARD> characters

Any word or character contained within greater than/less than symbols respresents a keyboard character.

b̸

The character b̸ indicates a blank or space character.

...

Ellipses indicate that a field entry may be repeated.

filespec

A field shown as filespec indicates a standard TRSDOS file specification of the form:
filename/ext.password:d(diskette name)

punctuation

Punctuation other than ellipses must be entered as shown.

delimiters

Commands must be separated from their operands by one or more blanks spaces. Multiple operands, where allowed, may be separated from each other by a comma, a comma followed by one or more blanks, or by one or more blanks. Blanks and commas may not appear within an operand.

<CTRL><character>

<CTRL><character> indicates the pressing and holding of the <CRTL> key and then pressing the key designated by <character>.

Introduction

General Description

BISYNC-80/3780 (the EMULATOR) is a communications protocol emulator that runs on the TRS-80 Model II microcomputer under control of the TRSDOS Operating System. The EMULATOR enables data transfer between a Model II microcomputer and a computer system equipped with bisynchronous hardware/software.

Since a computer system using bisynchronous hardware/software is usually IBM equipment and is located remotely to a mainframe computer system, this manual will refer to whatever the Model II is communicating with as the REMOTE.

BISYNC-80/3780 Functional Features and Options

When you connect your Model II by telephone to a REMOTE and run the EMULATOR, you can use EMULATOR commands to:

- Transmit a TRSDOS file or series of files to a REMOTE.
- Receive data from the REMOTE directly or indirectly into TRSDOS files. The files may be defined to contain one transmission from the REMOTE or as a series of files that will contain all transmissions from the REMOTE.
- Automatically (or operator selectable) translate ASCII/EBCDIC codes.
- Transmit and receive character coded (non-transparent) or binary (transparent) data.
- Automatically pad or truncate any records to match transmitted data to the requirements of the device protocol being used.
- Decode printer forms control (See Appendix B for a description of control codes and printer action.)
- Allow operator selection of Device Mode (2770,2780,3741, 3789, or CPU. CPU Mode is useful in microcomputer-to-microcomputer communications where

record padding and truncation are not needed).
- Enable interactive operation with interactive assignment of alternative command input from a disk file.
- Allow the interactive specification of IBM Remote Job Entry (RJE) commands on the Model II and RJE command replies displayed on the Model II.

Your Model II can communicate with the following REMOTE Devices:

- IBM S/360
- IBM S/370
- IBM 30-Series computers
- IBM 2780 Terminals
- IBM 3780 Terminals
- DEC PDP-11
- DEC VAX-11
- and other devices equipped with bisynchronous communications capabilites.

To use the EMULATOR, you'll need a TRS-80 Model II computer with 64K of RAM memory. You may include an optional line printer and additional floppy disk drives. Before the Model II can be used to communicate with a REMOTE, Serial Channel A on the Model II must be configured for synchronous communications. This must be done by a qualified Radio Shack service technician. Your local Radio Shack store or computer center will make these modifications for you.

Communications Line Requirements

The EMULATOR program may be operated at data rates of up to 19,200 baud with one of three communications links (see Table 1):

- Point-to-point (leased) or switched (dial-up) telephone lines with synchronous modems.
- Hard-wired connections with short-haul synchronous modems.
- Hard-wired connections with a synchronous modem eliminator.

Interface to the modem or modem eliminator is via an 11-conductor cable equipped with DB-25 type connectors on both ends. (See BISYNC-80/3780 Configuration and Appendix D for cable requirements.)

The EMULATOR communicates in half-duplex mode and requires only half-duplex communications facilities.

The EMULATOR may be run with a hard-wired connection and short-haul synchronous modems or a synchronous modem eliminator for communicating over short distances. The length of the communications path in these configurations is limited by the characteristics of the modems or by the modem eliminator used. The specifications of the modem or modem eliminator will show the baud rates and cable lengths that may be used with the device.

Modems are available from the telephone company (201-C and 208-B Modems). Modem eliminators are available from several companies.

Communications Links

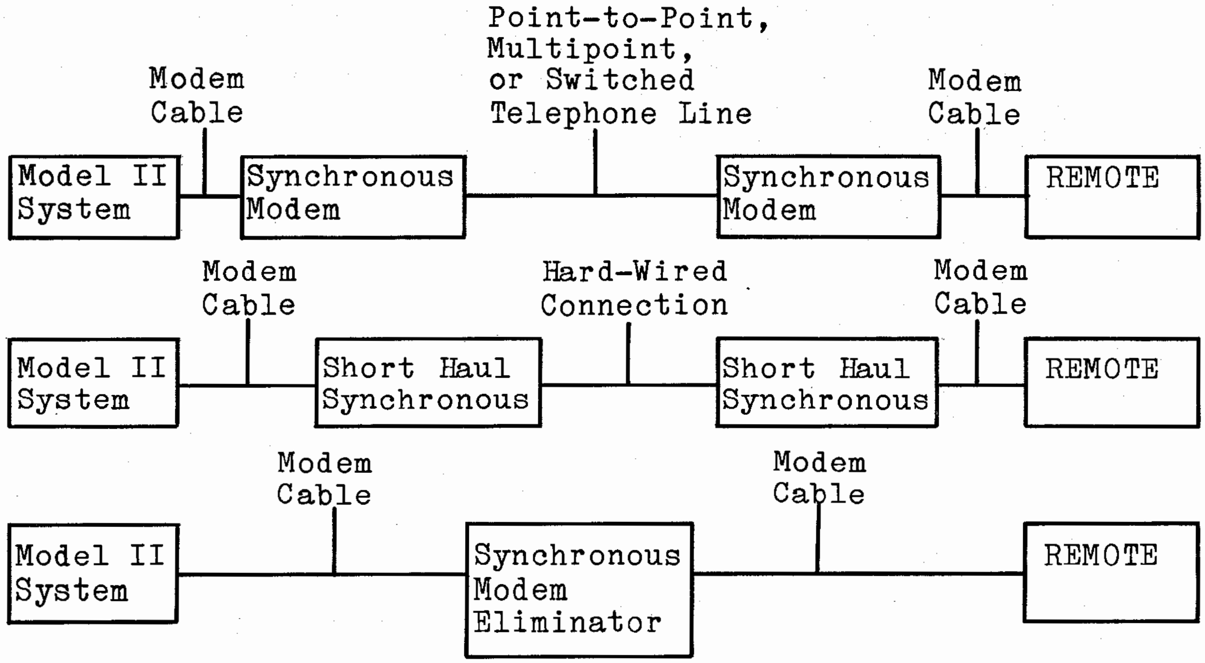


Table 1

REMOTE Requirements

Your Model II and the REMOTE must be equipped with compatible modems. Furthermore, the REMOTE must be capable of communicating with an IBM 2770, 2780, 3741, or 3780, or must be another microcomputer running the EMULATOR program.

Typical REMOTES include:

- TRS-80 Model II computers with BISYNC-80/3780 software.
- IBM 2770, 2780, 3741, and 3780 Terminals.
- IBM S/360*, S/370, and 30xx CPUs with Communications Controllers and IBM operating system RJE support.
- IBM 1130, 1800, SYSTEM/3, SYSTEM/7, and SYSTEM/34 Computers.
- DEC PDP-11 computers with the 2780 REMOTE Computer Systems software.
- DEC VAX-11 computers with the VAX-11 2780/3780 Protocol Emulator software.
- INTEL Microcomputer Development Systems with the Mainframe Link software.

(*) Except Model 44 and Model 67 in multiprocessor mode.

Note: The BISYNC-80/3780 program operates in point-to-point mode only, and cannot be used in a multipoint (polled) environment.

1/Using BISYNC-80/3780

Operational Overview

The EMULATOR enables the sharing of processing between a TRS-80 Model II system and a REMOTE processor. Two types of files may be transferred:

- **Text or character coded files**, such as Job Control Language (JCL), source, and listing files. These files contain only valid ASCII characters when stored at the Model II; they contain only EBCDIC characters when stored at the REMOTE.
- **Binary coded files**, such as object, link, library, and executable module files. These files contain data that cannot be represented by a valid ASCII or EBCDIC character.

The EMULATOR's SEND, RECEIVE, and PUNCH commands allow the transfer of both file types between the Model II and the REMOTE. The EMULATOR performs the ASCII/EBCDIC code translation, record padding, and line protocol handling so that a file does not logically change when transferred.

Transmission Types

Since the Binary Synchronous Communications (BSC) protocol defines some characters as line control characters, transmitting a file containing binary data will probably involve transmitting a data character that is also a line control character. To prevent this character from being interpreted as a line control character, a special transmission mode--Transparent Mode--must be used to transfer binary files. Transparent Mode may also be used for character coded files, but since Transparent Mode is less efficient in transferring data, this mode should be used only when required.

ASCII/EBCDIC Translation

Since data may be either binary (does not require code translation) or character coded (usually requires translation between ASCII and EBCDIC), the EMULATOR provides automatic translation for some files based on the file extension; the operator can also override this automatic translation by EMULATOR command.

In every command mode, files using the extensions /OBJ and /REL will be transmitted and received without ASCII/EBCDIC translation unless you specify translation is to take place. Filenames with other extensions will be translated unless you specify otherwise. Translation is not in effect in CPU device mode.

A list of file extension specifications for files not to be translated (/OBJ, /REL) can be altered to suit your needs. (See BISYNC-80/3780 Configuration for details.)

Translation is independent of the transparent mode selection. Files transmitted or received transparently may also be translated. Binary files containing data that is not a part of the ASCII character set cannot be translated and must be transmitted transparently.

Translation occurs on a file-by-file basis, so that each of a series of files transmitted (as a result of one command entered) has an individual translate/no-translate attribute. This method of transmisson simplifies the mixing of binary data that is not to be translated with character coded data that is to be translated (such as JCL).

When transmitting and translating data, ASCII form control characters (such as CR, LF, TAB, FF, etc.) which are contained in files are translated to their EBCDIC equivalents and are sent as a part of the data record containing the character unless the forms control character is also a record delimiter. ASCII line control characters (such as EOT, ETB, etc.) that may be imbedded in a record are also translated to their EBCDIC equivalents. Records containing line control characters must be transmitted transparently.

ASCII/EBCDIC translation tables are given in the User Configuration Source Module. They may be modified if your installation requires it. See BISYNC-80/3780 Configuration for details on changing the translate tables.

Record Padding and Truncation

Character coded data is maintained in your Model II as a string of ASCII records. On the other hand, the IBM terminals emulated are devices that use individual fixed-length card image and variable-length print line image EBCDIC records. When transmitting character coded files, the EMULATOR reformats (pads or truncates) file records into card images, translates the records to EBCDIC, and transmits them to the REMOTE. This reformatting is performed for all files that are translated from ASCII to EBCDIC and reformatting is not performed for files not translated. Translated character coded data received from the REMOTE is placed into a file or written to the line printer as a series of ASCII records.

Binary non-translated files are maintained in your Model II system as a series of binary records. When transmitting binary files, the EMULATOR obtains an 80-byte card image from a TRSDOS file and transmits it to the REMOTE without reformatting. Padding is not performed except on the last record transmitted from the file which may be padded with binary zeroes. Non-translated data from the REMOTE is placed into a file as a series of binary records without the insertion of record delimiters by the EMULATOR.

File Transmission Requirements Summary

File types that are commonly transferred by the EMULATOR are listed in Table 2. Each file type's translation, transparency, and padding attributes are also listed.

File Transmission Requirements Summary

		Transparent	Padding	
		Mode	Performed	
File-Type	Translated?	Required?	On Transmit?	Comments
JCL	Yes	No	Yes	
SOURCE	Yes	No	Yes	
LIST	Yes	No	Yes	
OBJECT	No	Yes	No	Last record padded
RELOCATABLE	No	Yes	No	Last record padded

Table 2

Starting Up

Before running the EMULATOR on a TRS-80 Model II, Serial Channel A must be free.

If you have not used Serial Channel A since TRSDOS was last initialized, you can proceed. If you have used Channel A, you must issue the TRSDOS Library Command **SETCOM A=OFF** to free up the channel. (See your TRS-80 Model II's owner's manual for details on SETCOM.)

When TRSDOS READY is displayed, you can start-up the EMULATOR by typing:

```
BIS3780 <ENTER>
```

The EMULATOR will begin execution and the start-up message, followed by the > prompt, will be displayed.

The EMULATOR is then running and ready to accept commands. You can then establish communications with the REMOTE.

Establishing the Telephone Data Link

If the Model II is using a point-to-point (dedicated) telephone line to communicate with the REMOTE, a telephone connection normally does not need to be established.

If the Model II is connected to a switched (dial-up) line, follow these steps:

1. Press the TALK button on the telephone and lift the telephone handset.
2. Dial the number of the REMOTE.
3. When the REMOTE answers and you hear a high-pitched tone, press the DATA button on the telephone and hang up the receiver. Your Model II is then connected to the REMOTE. If your modem has a Modem Ready (MR) light, the modem's MR light will illuminate when communications are established.

If the REMOTE number is busy, try again or use an alternate number.

If the REMOTE number does not answer, call a person at the REMOTE location and ask to have the line activated.

Data Transfer

When you're ready to send data to the REMOTE, use the SEND command, which must be in the following format: **SEND** filespec <ENTER> where filespec is a TRSDOS file or series of files. (SEND is described in detail later in this section.)

Before receiving data from a REMOTE, you must define a file for the data to go into or tell the EMULATOR to print the data (using a RECEIVE or PUNCH command).

To receive data to a file, type: PAGE 29

```
RECEIVE filespec <ENTER>
```

To receive data to the printer, type:

```
RECEIVE {PRT} <ENTER>
```

(RECEIVE and PUNCH are described in detail later in this manual.)

EMULATOR Commands

Command Syntax

Command syntax is a command's general form (like the grammar or structure of an English sentence). The syntax tells how to use keywords (such as RECEIVE, SEND, etc.) together with the necessary parameters and punctuation.

EMULATOR commands must have the following syntax:

COMMAND operand delimiter operand...

> COMMAND is the operation to be executed.
> operand is the operand(s) required by the command. operand must not contain commas or blank spaces.
> delimiter is a comma or blank space or a comma followed by one or more blank spaces.

Appendix F summarizes the EMULATOR's commands, abbreviations, and uses.

ABORT
Terminating an Operation

```
ABORT
```

Any operation in progress may be immediately ended by typing the command:

```
    ABORT <ENTER>
```

If communications are in progress, an EOT (End-of-Transmission) is sent to the REMOTE. The EMULATOR then closes all data files and returns to the idle state. This command may be used to terminate a transmission or reception at any time.

After ABORT is processed, the EMULATOR returns to normal processing. If the REMOTE was transmitting data to the Model II, the current receive file is closed and the RECEIVE or PUNCH command is considered satisfied jf the EMULATOR is not in Receive Chain mode.

Example

To ABORT a transmission in progress, type:

```
        ABORT <ENTER>
```

The current transmission will be immediately terminated.

CONSOLE
Alternate Command Input

CONSOLE *filespec*

CONSOLE may be abbreviated as C.
filespec is a standard TRSDOS file and specifies the name of the file (including any extension) to be assigned as the source of command input.

This command input assignment facility is provided by the EMULATOR to allow you to create predefined command files for executing frequently performed operations.

CONSOLE may be used to assign the command input to a disk file. The CONSOLE file may consist of any number of EMULATOR commands and must be created with the Editor/Assembler or the TRSDOS **BUILD** function. (Note: Parameter substitution in the command file is not available with this command.)

See Operation With CONSOLE and Submit Files later in this chapter for a complete discussion of EMULATOR operation when command input is assigned to a file, your TRS-80 Model II owner's manual for details on BUILD, and the Editor/Assembler manual.

Example

To assign console input to a file named EMULCN/SRC, type:

```
    CONSOLE EMULCN/SRC <ENTER>
```
or
```
    C EMULCN/SRC <ENTER>
```

The Console input device is then assigned to the specified file.

DEQUEUE

Deletion of Current RECEIVE/PUNCH

```
DEQUEUE remove

    DEQUEUE may be abbreviated as DEQ.
    remove is either REC or PUN.
        REC specifies that the current RECEIVE assignment
        be removed.
        PUN specifies that the current PUNCH assignment be
        removed.
```

The currently assigned RECEIVE or PUNCH file or device may be removed through use of the DEQUEUE command. For instance, you may need to disable (unassign) the printer to permit receiving data to a disk file. You may also want to change filenames when receiving in Receive Chain mode. DEQUEUE allows you to delete the current assignment so that a new one may be made.

If the EMULATOR is receiving data when DEQUEUE is entered, the command will not take effect until the data reception is complete.

Example

To eliminate the current RECEIVE assignment, type:

```
    DEQUEUE REC <ENTER>
or
    DEQ REC <ENTER>
```

To eliminate the current PUNCH assignment:

```
    DEQUEUE PUN <ENTER>
or
    DEQ PUN <ENTER>
```

EXIT
Return to TRSDOS

```
EXIT
```

When you need to return to TRSDOS (ending the EMULATOR), type:

```
EXIT <ENTER>
```

See Operation with CONSOLE and Submit Files later in this chapter for a discussion of EXIT when the Console is assigned to a file.

Example

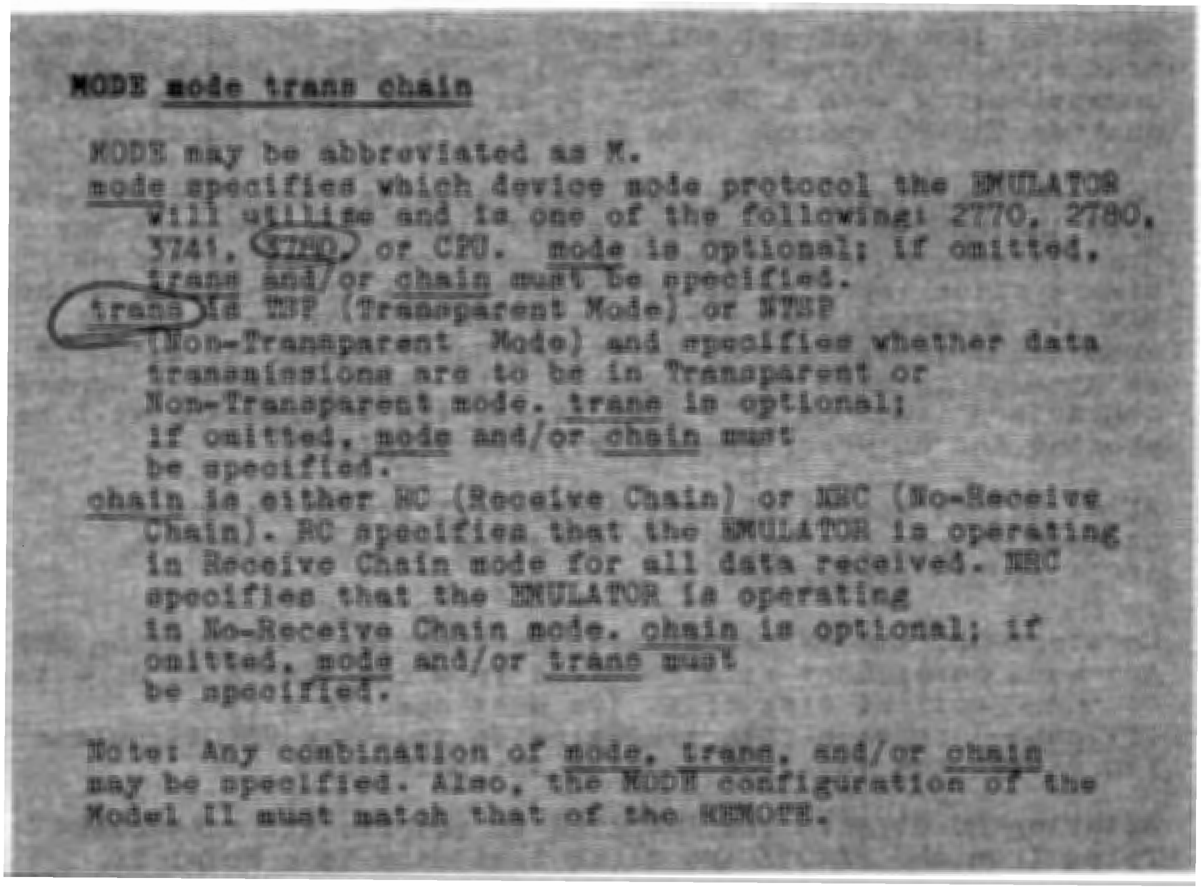
When you type:

```
EXIT <ENTER>
```

the Model II should display:

```
BIS-040 EXITED BISYNC-80/3780
TRSDOS READY
............................
```

MODE
Device Mode Change



MODE mode trans chain

MODE may be abbreviated as M.

mode specifies which device mode protocol the EMULATOR will utilise and is one of the following: 2770, 2780, 3741, 3780, or CPU. mode is optional; if omitted, trans and/or chain must be specified.

trans is TSP (Transparent Mode) or NTSP (Non-Transparent Mode) and specifies whether data transmissions are to be in Transparent or Non-Transparent mode. trans is optional; if omitted, mode and/or chain must be specified.

chain is either RC (Receive Chain) or NRC (No-Receive Chain). RC specifies that the EMULATOR is operating in Receive Chain mode for all data received. NRC specifies that the EMULATOR is operating in No-Receive Chain mode. chain is optional; if omitted, mode and/or trans must be specified.

Note: Any combination of mode, trans, and/or chain may be specified. Also, the MODE configuration of the Model II must match that of the REMOTE.

MODE allows the options used with the transmission and reception Modes to be changed. Those options are:

- Device Mode (2770, 2780, 3741, 3780 or CPU-CPU line Protocol)
- Transparent or Non-Transparent transmission mode
- Receive File chaining or No-Receive File chaining

Since each of the Device Modes use a unique protocol, it is necessary that the EMULATOR be in the correct Device Mode before any communication with the REMOTE begins. It is also essential that the Model II Device and Transparency Mode selections match that of the REMOTE for efficient data tranfer.

On start-up, the Device Mode is specified in the User Configuration Module. The User Configuration Module, as delivered, specifies the start-up Device Mode as 3780. Entering 2770, 2780, 3741, or CPU directs the EMULATOR to utilize a specific line protocol and record blocking. CPU (CPU-CPU mode) transmits transparent records only. CPU should be used only when communicating with a DEC PDP-11, an INTEL Development System, or another TRS-80 system. (See BISYNC-80/3780 Configuration.)

On start-up, trans is set at NTSP; however, transparent data from the REMOTE is accepted at any time, regardless of the Transparency Mode setting. trans must be specified before binary files (such as object modules) are transmitted to the REMOTE. Entering NTSP directs the EMULATOR to perform all subsequent transmissions utilizing Non-Transparent mode. If an attempt is made to transmit a file containing line control characters (such as EOT, ETB, etc.), the transmisson will be aborted and an error message will be displayed.

After TSP is specified, all subsequent transmissions will utilize Transparent Mode. This option has no effect on ASCII/EBCDIC translation.

On start-up, the EMULATOR uses the chaining mode setting NRC and each subsequent file received from the REMOTE is directed to the specified disk file. You must specify a file (filespec) on each file transmission from the REMOTE unless you direct the received data to the printer.

Entering RC directs the EMULATOR to receive all subsequent files from the REMOTE and store that data in a Model II TRSDOS file on disk. This is Receive Chaining. When Receive Chaining is in effect, the filename extension specified with RECEIVE or PUNCH is ignored. The first file received from the REMOTE will be named filename/001; the second file will be filename/002; etc. RC is overridden by the assignment of the command line input to a disk file, for the duration of the assignment.

Example

To change the EMULATOR to 2770 Mode, type:

```
    MODE 2770 <ENTER>
```

or

```
    M 2770 <ENTER>
```

To change the EMULATOR to Transparent Mode, type:

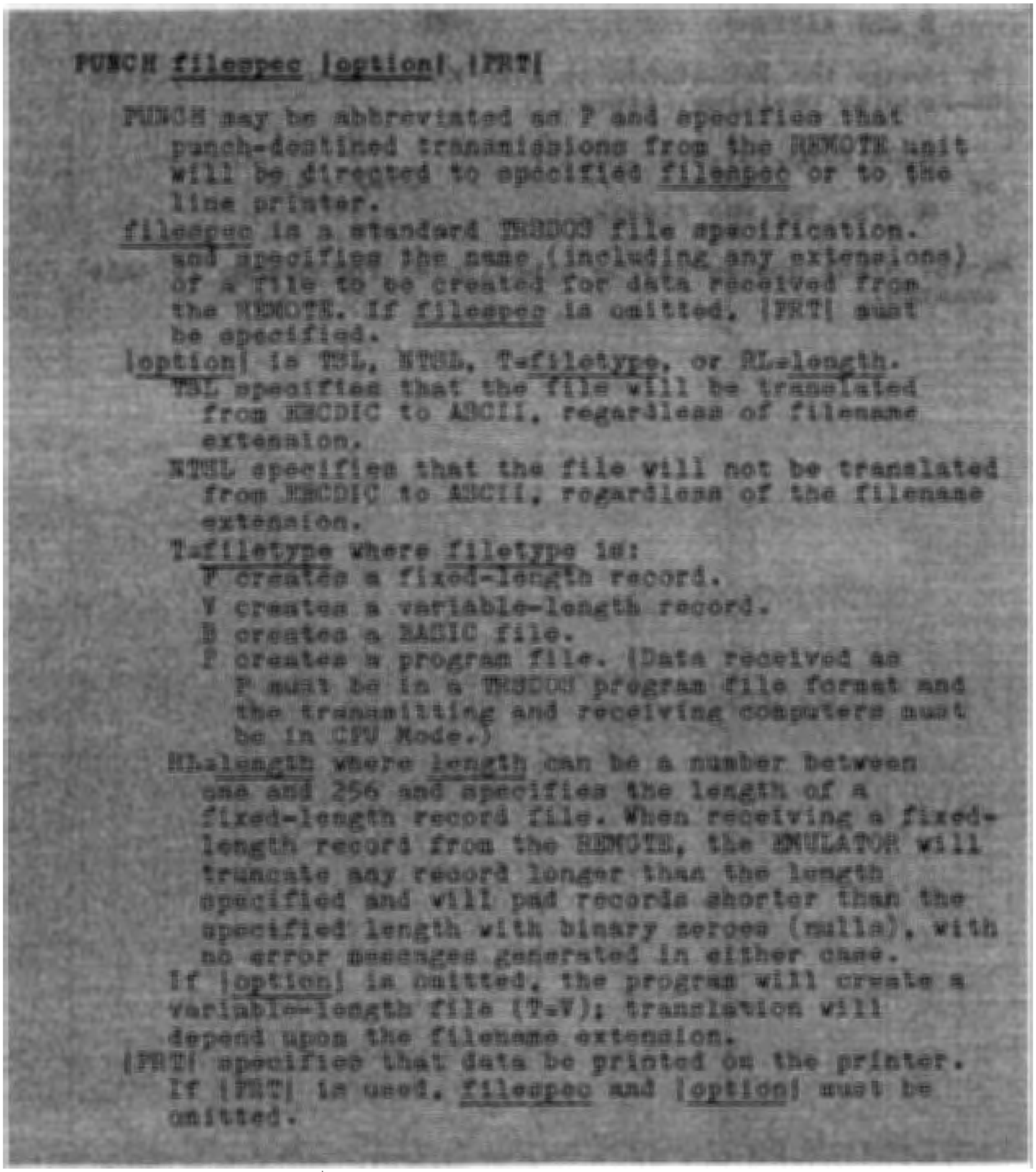
```
    MODE TSP <ENTER>
or
    M TSP <ENTER>
```

To change the EMULATOR's to 2780 Mode, Transparent, with No-Receive Chaining, type:

```
    MODE 2780,TSP,NRC <ENTER>
or
    M 2780,TSP,NRC <ENTER>
```

Note: Blank spaces may be substituted for commas in this example.

PUNCH
Receive Data



PUNCH filespec |option| |PRT|

PUNCH may be abbreviated as P and specifies that punch-destined transmissions from the REMOTE unit will be directed to specified filespec or to the line printer.

filespec is a standard TRSDOS file specification. and specifies the name (including any extensions) of a file to be created for data received from the REMOTE. If filespec is omitted, |PRT| must be specified.

|option| is TSL, NTSL, T=filetype, or RL=length.

TSL specifies that the file will be translated from EBCDIC to ASCII, regardless of filename extension.

NTSL specifies that the file will not be translated from EBCDIC to ASCII, regardless of the filename extension.

T=filetype where filetype is:

F creates a fixed-length record.
V creates a variable-length record.
B creates a BASIC file.
P creates a program file. (Data received as P must be in a TRSDOS program file format and the transmitting and receiving computers must be in CPU Mode.)

RL=length where length can be a number between one and 256 and specifies the length of a fixed-length record file. When receiving a fixed-length record from the REMOTE, the EMULATOR will truncate any record longer than the length specified and will pad records shorter than the specified length with binary zeroes (nulls), with no error messages generated in either case.

If |option| is omitted, the program will create a variable-length file (T=V); translation will depend upon the filename extension.

|PRT| specifies that data be printed on the printer. If |PRT| is used, filespec and |option| must be omitted.

Note: If T=F is used, RL=length must be used. If T=V, T=B, or T=F is used, RL=length cannot be used. (T=B is the same as T=F,RL=1).

Punch data received from the REMOTE (except RJE console command responses) will be printed on the line printer or written in a file specified by the PUNCH command. If the REMOTE begins transmitting PUNCH data and a printer or file has not been defined for the transmission, the EMULATOR will display a request for a PUNCH file definition. The EMULATOR will then wait until the file is defined.

If Receive Chaining (RC) is in effect, all transmissions from the REMOTE will be directed to the specified filespec. If No-Receive Chaining (NRC) is in effect, only the next transmission from the REMOTE will be directed to filespec; after receiving data, a new PUNCH command must be issued. (Note: Receive Chaining is in effect if you specify {PRT}).

You may begin numbering Receive Chaining file extensions at any numbered extension by specifying a three digit numeric extension in the PUNCH command. The EMULATOR will add one to the extension specified and use the result as the first extension to be created. If, for instance, you specify the filename TEST/000 and the EMULATOR is in RC Mode, the first filename used will be TEST/001.

If any filename exists prior to the PUNCH command, you will be given the opportunity to specify that the existing file is to be deleted and overwritten.

Example

If punch-destined data is also to be directed to the printer, the command:

```
    PUNCH {PRT} <ENTER>
or
    P {PRT} <ENTER>
```

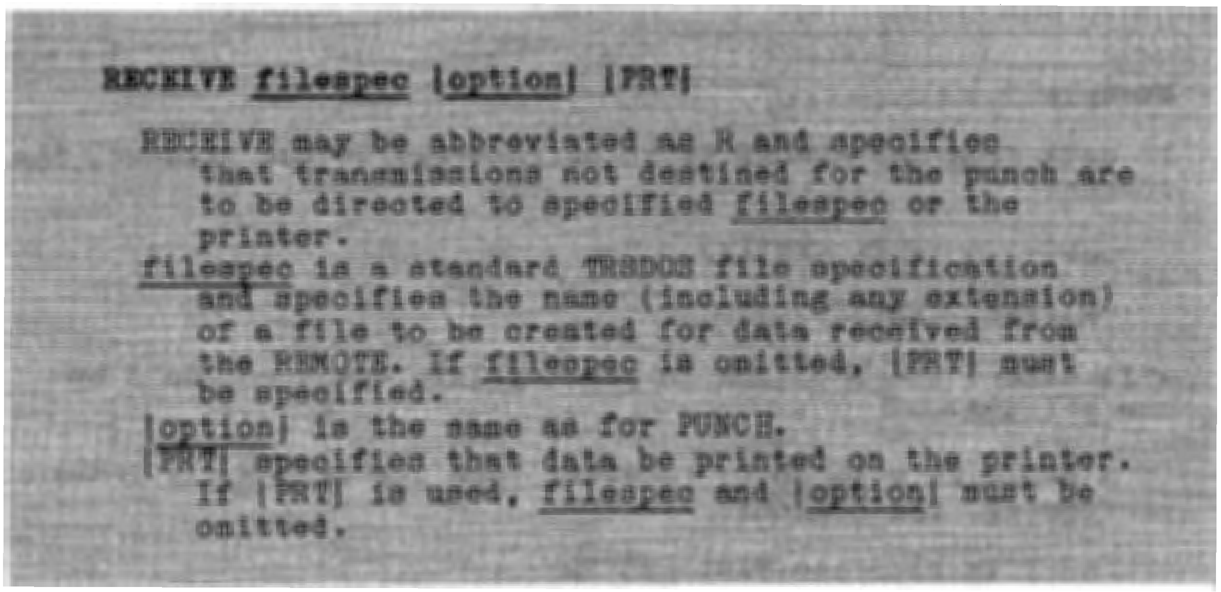
will direct the EMULATOR to place punch-destined data to the printer.

If punch-destined fixed-length binary file is to be received to a file, the commands:

```
    PUNCH FILE/BIN{NTSL,T=F,RL=80} <ENTER>
or
    P FILE/BIN{NTSL,T=F,RL=80} <ENTER>
```

will direct the EMULATOR to place the punch data into the file name FILE/BIN. The specification NTSL (No-Translate) is required since the file extension (if it is not /OBJ or /REL) specifies file translation from EBCDIC to ASCII by default.

RECEIVE
Receive Data



```
RECEIVE filespec {option} {PRT}

   RECEIVE may be abbreviated as R and specifies
      that transmissions not destined for the punch are
      to be directed to specified filespec or the
      printer.
   filespec is a standard TRSDOS file specification
      and specifies the name (including any extension)
      of a file to be created for data received from
      the REMOTE. If filespec is omitted, {PRT} must
      be specified.
   {option} is the same as for PUNCH.
   {PRT} specifies that data be printed on the printer.
      If {PRT} is used, filespec and {option} must be
      omitted.
```

All printer destined data received from the REMOTE (except RJE console command responses and punch data) will be printed on the line printer or written in a file specified by the RECEIVE command. If the REMOTE begins transmitting RECEIVE data and a printer or file has not been defined for the transmission, the EMULATOR will display a request for a RECEIVE file definition. The EMULATOR will then wait until the file is defined.

If Receive Chaining (RC Mode) is in effect, all transmissions from the REMOTE will be directed to the specified filespec. If No-Receive Chaining is in effect (NRC), only the next transmission from the REMOTE will be directed to filespec; after receiving data, a new RECEIVE command must be issued. (Note: Receive Chaining is used if you specify {PRT}).

You may begin numbering Receive Chaining file extensions at any numbered extension by specifying a three digit numeric extension in the RECEIVE command. The EMULATOR will add one to the extension specified and use the result as the first extension to be created. If, for instance, you specify the filename TEST/000 and the EMULATOR is in RC mode, the first filename used will be TEST/001.

If any filename exists prior to the RECEIVE command, you will be given the opportunity to specify that the existing file is to be deleted and overwritten.

Example

If data from the REMOTE is to be received to the printer, the command:

```
    RECEIVE {PRT} <ENTER>
or
    R {PRT} <ENTER>
```

will direct the EMULATOR to place printer-destined data on the printer.

If a BASIC file is to be received from the REMOTE for later processing, the command:

```
    RECEIVE FILE/DAT{T=B} <ENTER>
or
    R FILE/DAT{T=B} <ENTER>
```

will direct the EMULATOR to place the next printer destined transmission from the REMOTE into a file named FILE/DAT, formatted for BASIC processing. If the REMOTE transmits the file to the printer, the file will be placed in FILE/DAT by the EMULATOR.

If a binary file is to be received from the REMOTE for later processing, the command:

```
    RECEIVE FILE/BIN{NTSL} <ENTER>
or
    R FILE/BIN{NTSL} <ENTER>
```

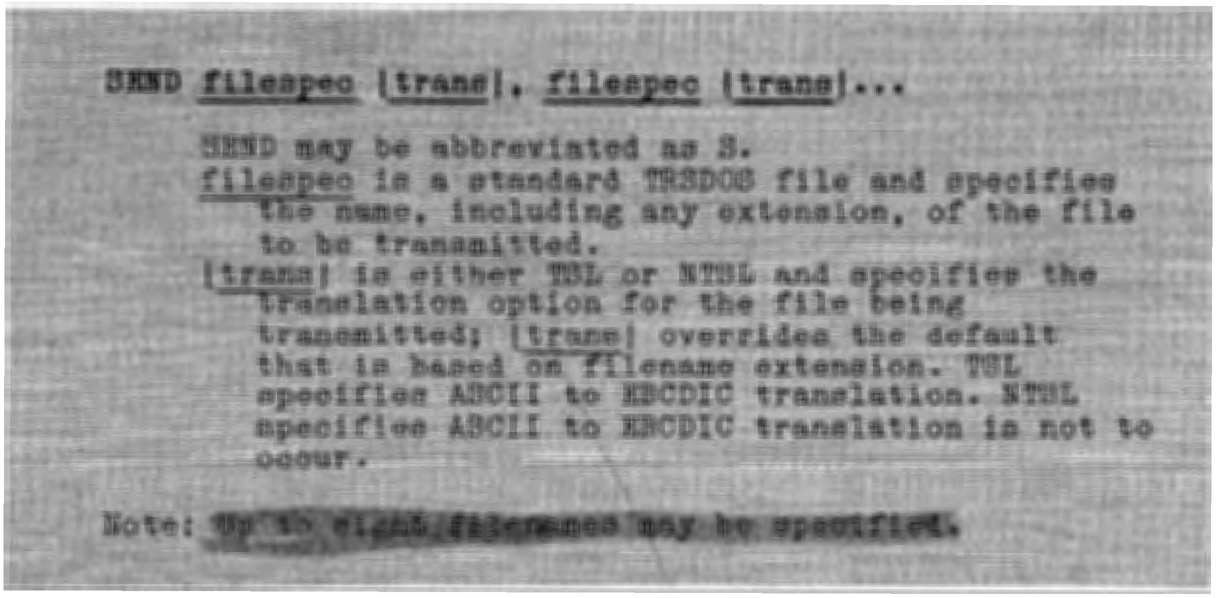
will direct the EMULATOR to place the next printer-destined transmission from the REMOTE into the file named FILE/BIN. The specification NTSL (No-Translate) is required since the file extension (if it is not /OBJ or /REL) specifies file translation from EBCDIC to ASCII by default.

If a printer-destined text file is received to the printer and a punch-destined fixed length binary file is to be received to a file, the commands:

```
    RECEIVE {PRT} <ENTER>
or
    R {PRT} <ENTER>
```

will direct the program to place the files as needed.

SEND
Data File Transmission



```
SEND filespec [trans], filespec [trans]...

     SEND may be abbreviated as S.
     filespec is a standard TRSDOS file and specifies
        the name, including any extension, of the file
        to be transmitted.
     [trans] is either TSL or NTSL and specifies the
        translation option for the file being
        transmitted; [trans] overrides the default
        that is based on filename extension. TSL
        specifies ASCII to EBCDIC translation. NTSL
        specifies ASCII to EBCDIC translation is not to
        occur.

Note: Up to eight filenames may be specified.
```

Data files may be transmitted to the REMOTE with or without ASCII to EBCDIC translation and in Transparent or Non-Transparent Mode. The translation selection may be automatic, depending on filename extension, or explicitly directed by the SEND command. Transparency is selected by the MODE command.

Multiple filenames may be specified to cause logical concatenation of the files and transmission of the files as one data set. Up to eight filenames may be specified.

When multiple files are transmitted, the translation attribute based on filename extension or command option specified is on a file-by-file basis, and it is possible to mix translate and non-translate files in one transmission.

This mix is needed when submitting a binary file to an IBM REMOTE, since the binary file must be surrounded by IBM Job Control Language (JCL) statements that direct the processing of the binary file at the REMOTE. The Job Control Language may be a standard ASCII character coded file. This file should be translated and padded by the EMULATOR; the binary file transmitted should not.

Example

To transmit a Job Control Language (JCL) file named JOB/JCL followed by a binary file named FILE/OBJ (in Transparent Mode), type:

```
    MODE TSP <ENTER>
    SEND JOB/JCL,FILE/OBJ <ENTER>
```

To transmit a JCL file named JOB1/JCL followed by a character coded (text) file named TEXT/FIL, type:

```
    SEND JOB1/JCL,TEXT/FIL <ENTER>
or
    S JOB1/JCL,TEXT/FIL <ENTER>
```

To transmit a JCL file named JOB2/JCL followed by a binary file named BIN/FIL (in Transparent Mode), type:

```
    MODE TSP <ENTER>
    SEND JOB2/JCL,BIN/FIL{NTSL} <ENTER>
or
    S JOB2/JCL,BIN/FIL{NTSL} <ENTER>
```

Note that the binary file BIN/FIL would be ASCII/EBCDIC translated if the No-Translate option were omitted from the command. This is because the filename extension (/FIL) does not specify that translation is to be bypassed.

STAT
Status Inquiry

```
     STAT
```

The EMULATOR's Status Inquiry command may be used to obtain the current EMULATOR status. STAT causes the Model II to display:

- Current transmission mode (transmitting, receiving, or idle).
- Current mode set options.
- Current RECEIVE/PUNCH device or filenames (if any).
- Block counts and error retry counts for the entire current EMULATOR execution and for the current operation in progress. This includes the:
  - number of data blocks transmitted successfully.
  - number of data blocks received successfully.
  - number of NAKs received in response to transmitted blocks, indicating the number of errors detected in blocks received by the REMOTE.
  - number of abnormal or timeout responses to transmitted blocks (number of ENQs sent requesting response). indicating the number of blocks that were not received by the REMOTE because of communications errors.
  - number of NAKs sent in response to received blocks indicating the number of errors detected in blocks transmitted by the REMOTE.
  - number of ENQs received requesting a response to a received block, indicating the number of blocks that were not received by the EMULATOR because of errors in the communications line.

The block and error retry counts give a good indication of the quality of the communications (telephone) line. If the error counts are greater than approximately 1% of the block counts and the communications line is short (less than 500 miles), then the communications link is not performing correctly and end-to-end modem tests should be run with the modems being used. Instructions provided with most modems describe how to run these tests.

Example

Typing:

```
STAT <ENTER>
```

will produce a display similar to:

```
BIS-016  CURRENT SYSTEM STATUS
SYSTEM IS RECEIVING, IN 2780 DEVICE MODE
TRANSPARENCY IS NOT IN EFFECT (NTSP MODE)
RECEIVE CHAINING IS NOT IN EFFECT (NRC MODE)
CURRENT FILES ASSIGNED:
RECEIVE FILE- {PRT}
PUNCH FILE  - PUNCH/OBJ
TRANSMISSION LINE STATISTICS:
                SENT           RECEIVED
TOTAL BLOCKS:   0385            0184
TOTAL NAKS:     0000            0001
TOTAL ENQS:     0001            0000
CURRENT BLOCKS: 0000            0018
CURRENT NAKS:   0000            0000
CURRENT ENQS:   0000            0000
***END OF SYSTEM STATISTICS***
```

Operation With CONSOLE and Submit Files

When the command input is assigned to a file by the EMULATOR's CONSOLE command, special processing factors must be taken into consideration.

All commands read are executed immediately, if possible. Since it may not be possible to immediately execute a SEND, RECEIVE, or PUNCH command, these commands are read and queued for later execution. Each queue (SEND, RECEIVE, and PUNCH) will hold two commands, so that two SEND, RECEIVE, and PUNCH commands may be read and queued. When the third unexecuted SEND, RECEIVE, or PUNCH command is read, command file input operations are suspended until the corresponding queue is completed (by a prior command) and reduced by one. At that time, command file input operations resume. The DEQUEUE command may also be used to dequeue a RECEIVE or PUNCH file name and cause command file input operations to resume.

If the EXIT command is read from a file, the EMULATOR's EXIT command is not processed until all queues are empty and all commands have been satisfied. Note: The EMULATOR will process an EXIT command from the Model II keyboard immediately.

At end-of-file on the command input file, the command input assignment reverts to the Model II keyboard, regardless of how the assignment was made.

Nesting of CONSOLE commands (e.g., having a CONSOLE command within a console file) causes a switch of the console input to the new file specified. At end-of-file on the new file, however, console assignment reverts to the keyboard, not to the original file.

Entering a <CTRL><E> on the Model II keyboard switches command input from a console file assignment to the keyboard. Similarly, a <CTRL><E> in the file assigned switches command input from the file to the keyboard. The console input assignment reverts to the file when a second <CTRL><E> is entered on the keyboard.

To summarize, when the console is assigned to a file:

- Receive chaining is not in effect.
- Commands are queued to a depth of two.

- Command input is suspended if there is no room in queue for commands.
- The EXIT command is not processed until all outstanding commands have been processed.
- End-of-file reverts console control to the keyboard.
- <CTRL><E> switches command input back and forth between the file and the keyboard.
- A CONSOLE command can appear in a console input file, but console control will not automatically revert to the file containing the command.
- Commands are processed immediately, if possible.

Example

If you have created a command file named COMMAND/FIL which contains the commands:

```
    MODE    RC
    RECEIVE {PRT}
    PUNCH   PUNCH/FIL
    SEND    FILE1/JCL,FILE2/TXT
```

and you issue the console asignment command:

```
    CONSOLE COMMAND/FIL <ENTER>
```

or

```
    C COMMAND/FIL <ENTER>
```

the EMULATOR will:

- Read and process the MODE command, setting Receive Chain mode.
- Read the RECEIVE and PUNCH commands and set the RECEIVE device to the line printer and the PUNCH device to the PUNCH/FIL.
- Send the files FILE1/JCL and FILE2/TXT.

Note that the SEND command will probably be processed before any data is received from the REMOTE, since the SEND command can be processed immediately and the RECEIVE and PUNCH commands must wait for the REMOTE to transmit.

RJE (REMOTE Job Entry) Commands

When running the EMULATOR in 2780 and 3780 Device Mode, the EMULATOR supports an RJE console, using the Model II. This facility allows entering any HASP, JES, POWER, or RES console command which is supported by the REMOTE and receiving the response from the REMOTE back to the Model II.

To be interpreted as an RJE console command, the command string must begin with:

- A dollar sign ($) (if the command is a HASP or JES).
- An asterisk (*) (if the command is a POWER or RES).

The EMULATOR replaces the dollar sign with the HASP string /*$ or the asterisk with the POWER string *b̸...b̸. It then appends the command and transmits the result to the REMOTE.

The next transmission from the REMOTE is assumed to be the reply to the command and is displayed on the Model II. Special RJE command identifiers may be used to send commands that do not cause the REMOTE to reply. (See BISYNC-80/3780 Configuration for a description of these identifiers.)

The RJE command feature is particularly useful in displaying and changing the status of jobs submitted to the REMOTE for processing. The ability to transmit commands from a TRS-80 and receive data back for display prevents having to exit the EMULATOR to create or display files.

The dollar sign and asterisk values and their related string replacements are in the User Configuration Module and may be modified to be any user chosen values. (See BISYNC-80/3780 Configuration for a description of the modification process.)

An RJE command entered or read from a command input file while data is being received from a REMOTE will cause interruption of the REMOTE's transmission. The command will be processed and the REMOTE transmission to resume after the command reply is received. This interrupt feature allows the purging of unwanted data from the REMOTE. A current RECEIVE or PUNCH file is closed when the interrupt is processed. A new RECEIVE or PUNCH command must be issued to contain the continuation of the data if it is not purged and if the EMULATOR is not in Receive Chaining Mode.

Unsolicited messages from the REMOTE to the Model II (such as REMOTE broadcast messages) are treated as data and will be placed on the printer or in the current receive file.

Example

To display the status of a REMOTE HASP system's input and output queues, type:

```
    $DQ <ENTER>
```

The command is sent to the REMOTE and the display of queue status returned by the REMOTE is displayed on the Model II.

To purge a job running on a remote POWER system (with the job identified as JOB 123), type:

```
   *PJ123 <ENTER>
```

The job will then be purged. The confirmation message from the REMOTE is displayed on the Model II.

Program Messages

The EMULATOR issues messages that inform you of current execution status, including the start and end of any data transmission operation. Error conditions that result in aborted transmissions are reported, but transient correctable errors are not. All messages are of the form:

BIS-*nnn* *message*

nnn is a unique numeric message identifier that provides easy reference to the message explanation.

message is a short message briefly describing the action required, condition found, or program status.

EMULATOR messages are documented in detail in Appendix A.

2/Sample Session

The following sample session shows what is necessary to send a job to a REMOTE from a Model II, receive back a listing (on the line printer), and a file (to be processed by a BASIC program).

Note: In the following sample session, the command line input data (e.g., what you type in) is followed by <ENTER>; all EMULATOR messages (e.g., what is displayed on the Model II) are not followed by <ENTER>. Line numbers are for reference purposes only and do not actually appear on the Model II display. An explanation (referenced by individual line numbers) follows the sample session.

```
0001 TRSDOS READY
0002 SETCOM A=OFF <ENTER>
0003 TRSDOS READY
0004 BIS3780 <ENTER>
0005 BISYNC-80/3780, COMMUNICATIONS EMULATOR V3.4
0006 (C) COPYRIGHT 1981 MICRO-INTEGRATION, INC.
0007 LICENSED TO TANDY CORPORATION
0008 >R {PRT} <ENTER>
0009 >P BASIC/FIL(T=B) <ENTER>
0010 >S JCL/SND,DATA/SND <ENTER>
0011 >
0012 BIS-002 TRANSMITTING
0013 >
0014 BIS-004 TRANSMISSION COMPLETE
0015 >STAT  <ENTER>
0016 SYSTEM STATUS
0017 SYSTEM IS IDLE, IN 3780 DEVICE MODE
0018 TRANSPARENCY IS NOT IN EFFECT (NTSP MODE)
0019 RECEIVE CHAINING IS IN EFFECT (RC MODE)
0020 CURRENT FILES ASSIGNED:
0021 RECEIVE FILE - {PRT}
0022 PUNCH FILE   - BASIC/001
0023 TRANSMISSION LINE STATISTICS
0024                        SENT      RECEIVED
0025 TOTAL BLOCKS:          0023      0000
0026 TOTAL NAKS:            0000      0000
0027 TOTAL ENQS:            0000      0000
0028 CURRENT BLOCKS:        0023      0000
```

```
0029 CURRENT NAKS:           0000      0000
0030 CURRENT ENQS:           0000      0000
0031 ***END OF STATISTICS DISPLAY***
0032 >
0033 BIS-003 RECEIVING
0034 >
0035 BIS-005 MESSAGE RECEIVED COMPLETE
0036 >
0037 BIS-003 RECEIVING
0038 >
0039 BIS-005 MESSAGE RECEIVED COMPLETE
0040 STAT <ENTER>
0041 SYSTEM STATUS
0042 SYSTEM IS IDLE, IN 3780 DEVICE MODE
0043 TRANSPARENCY IS NOT IN EFFECT (NTSP MODE)
0044 RECEIVE CHAINING IS IN EFFECT (RC MODE)
0045 CURRENT FILES ASSIGNED:
0046 RECEIVE FILE - {PRT}
0047 PUNCH FILE   - BASIC/002
0048 TRANSMISSION LINE STATISTICS
0049                        SENT     RECEIVED
0050 TOTAL BLOCKS:          0023     0210
0051 TOTAL NAKS:            0001     0000
0052 TOTAL ENQS:            0000     0000
0053 CURRENT BLOCKS:        0000     0191
0054 CURRENT NAKS:          0000     0000
0055 CURRENT ENQS:          0000     0000
0056 ***END OF STATISTICS DISPLAY***
0057 >EXIT <ENTER>
0058 BIS-040 EXITED BISYNC-80/3780
0059 TRSDOS READY
```

Explanation of Model II Display

Line Number	Explanation
0001	TRSDOS READY is displayed, showing that TRSDOS is running and ready for a command.
0002	The TRSDOS SETCOM command is entered to free Serial Channel A. (If Serial Channel A has not been used prior to this step, lines 2 and 3.)
0003	TRSDOS READY is displayed, showing that TRSDOS is running and ready for a command.
0004	The BISYNC-80/3780 start-up command is entered.
0005-0007	The BISYNC-80/3780 start-up message is displayed.
0008	A RECEIVE command directing printer-destined data to the line printer is entered.
0009	A PUNCH command directing punch-destined data to a file named BASIC/FIL is entered. Note that the file-type is specified as T=B; this creates a BASIC file.
0010	A SEND command is issued; transmission of the files named JCL/SND and DATA/SND to the REMOTE begins.
0011	The BISYNC-80/3780 command prompt is displayed showing that you can enter a command at this point.
0012	The TRANSMITTING message indicates that the previous SEND command is being executed and data is being sent to the REMOTE.

0013 The command prompt is displayed.

0014 The TRANSMISSION COMPLETE message indicates that the previous SEND command has completed normally.

0015 A STATUS command is entered to get the current Model II EMULATOR status.

0016-0031 The current Model II EMULATOR status is displayed. (Note that the punch file assigned is named BASIC/001. The /001 extension is used instead of the /FIL extension specified in the command; this is because the EMULATOR is in Receive Chain Mode.)

0032 The command prompt is displayed.

0033 The RECEIVING message indicates that the REMOTE is transmitting data to the EMULATOR. If the REMOTE is transmitting printer-destined data, that data will begin printing on the line printer at this time. If the REMOTE is transmitting punch-destined data, that data will be put into the file defined by the PUNCH command.

0034 The command prompt is displayed.

0035 The MESSAGE RECEIVED COMPLETE indicates that the transmission from the REMOTE has completed normally.

0036 The command prompt is displayed.

0037 The RECEIVING message indicates that the REMOTE is transmitting a second time to the EMULATOR. If the data is destined for the printer, it will be printed. If the data is destined for the punch, it will be placed in the punch file.

0038 The command prompt is displayed.

0039 The MESSAGE RECEIVED COMPLETE indicates that the transmission from the REMOTE has completed normally.

0040 A STATUS command is entered to display the current SYSTEM STATUS. (Note that the punch file extension is now /002--one of the two files received previously must have been directed to the punch. Also note that the block counts for data transmitted and received have been incremented to show the number of blocks that were sent and received.)

0057 The EXIT command is entered.

0058 The EMULATOR EXIT message is displayed.

0059 TRSDOS READY is displayed.

3/EMULATOR Diagnostics

Two powerful diagnostic tools are available in the EMULATOR:

- A loopback test that can verify the proper operation of the complete Model II system (computer, cables, and modem).
- A diagnostic dump facility that can help diagnose line protocol problems.

ALTEST
Analog Loopback Test

ALTEST

The EMULATOR may be directed to run an Analog Loopback Test (ALT) by typing:

```
    ALTEST <ENTER>
```

This command directs the transmission and the simultaneous reception of data blocks by the Model II. When an image of the data transmitted is received back by the EMULATOR, it is compared to the transmitted data. If the data is the same, the Model II and other components (modem, cables, etc.) in the data path are working correctly. If data is not received back or if the data is not the same, some portion of the data path is not working.

The Analog Loopback Test can be accomplished if the modem is equipped with the Analog Loopback feature. This configuration tests the entire local data path, including the modem.

To run the Analog Loopback Test, put the modem into analog loopback (usually you must push the AL button on the modem) and type **ALTEST** <ENTER>. The EMULATOR will transmit data blocks and will simultaneously receive back the data sent, comparing the two to insure a match.

Once started, the test sequence is repeated continuously until any character is entered on the keyboard or until an error occurs.

If an error occurs, a descriptive message will be displayed on the Model II display and the test will halt. If the error indicates a mismatch of transmitted/received data, the transmitted and received characters will be displayed in the error message. Facilities that show errors while running the Analog Loopback Test must be repaired before they may be used for remote communications.

Table 3 shows the error messages that can be produced and lists the probable cause of each error.

If a character is typed into the Model II and no errors have occurred, the EMULATOR will write a BIS-022 message to the Model II display, indicating successful completion, and terminate the test.

Example

Assuming the modem is in analog loopback, type:

```
    ALTEST <ENTER>
```

The EMULATOR will then issue the message:

```
    BIS-018 LOOPBACK TEST
            ENTER ANY CHARACTER TO TERMINATE TEST
```

If no errors are indicated after the EMULATOR has been running for a period of time, you must enter any keyboard character (or just press <ENTER>) to terminate the test. The EMULATOR will issue the message:

```
    BIS-022 LOOPBACK TEST COMPLETED WITHOUT ERROR
```

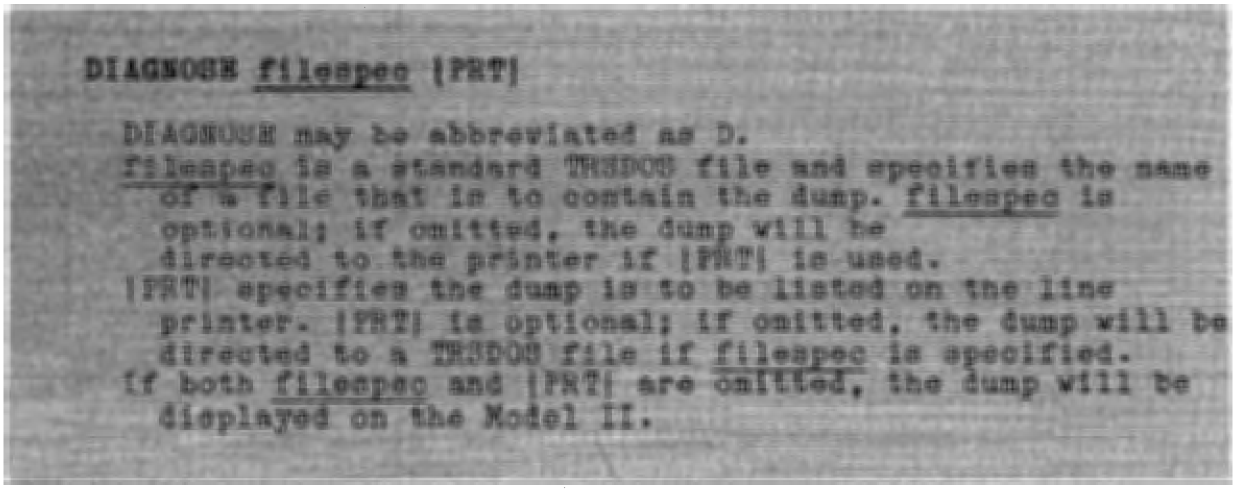
If an error is reported, the cause of the error must be found and corrected before remote communications can be attempted.

Loopback Errors and Possible Causes

Error Number	Error Message	Possible Cause
BIS-023	INVALID CHANNEL STATUS	Serial Channel A is defective.
BIS-024	TRANSMITTER TIMEOUT	The modem cable is not connected. The modem is not in Analog Loopback. The modem cable or modem is defective. Serial Channel A is defective.
BIS-020	NO DATA PATH ESTABLISHED	The modem is not in Analog Loopback. The modem cable or modem is defective. Serial Channel A is defective.
BIS-021	DATA PATH ERROR	The modem cable or modem is defective. Serial Channel A is defective.

Table 3

DIAGNOSE
Diagnostics



DIAGNOSE filespec |PRT|

DIAGNOSE may be abbreviated as D.
filespec is a standard TRSDOS file and specifies the name of a file that is to contain the dump. filespec is optional; if omitted, the dump will be directed to the printer if |PRT| is used.
|PRT| specifies the dump is to be listed on the line printer. |PRT| is optional; if omitted, the dump will be directed to a TRSDOS file if filespec is specified.
If both filespec and |PRT| are omitted, the dump will be displayed on the Model II.

The EMULATOR contains a circular buffer that logs all communications line activity. This buffer may be formatted and listed on the Model II display, the line printer, or put into a disk file.

DIAGNOSE produces a hexadecimal display of the line activity that is recorded in the buffer (see Table 4). Data transmitted to the line is preceded by the *****DATA TRANSMITTED***** identifier; data received from the line is preceded by the *****DATA RECEIVED***** identifier.

DIAGNOSE is intended to allow support personnel to examine a record of line activity in the unlikely event line protocol problems occur. The interpretation of traced activity is a complex task that is beyond the scope of this manual.

DIAGNOSE suspends all EMULATOR activity for the duration of the processing of the command. For this reason, DIAGNOSE should be issued only when suspected line protocol problems are encountered.

The DIAGNOSE display can be paused by pressing <HOLD>. The display can be resumed by pressing <HOLD> again. The DIAGNOSE display can be ended by pressing <BREAK>.

```
=====================================================
          Sample DIAGNOSE Display
-------------------------------------------------------
BIS-031  DIAGNOSTIC DUMP

***TRANSMITTED DATA***
PAD SYN SYN SYN SYN ENQ PAD PAD

*** RECEIVED DATA***
DLE AK0 PAD

*** TRANSMITTED DATA***
PAD SYN SYN SYN SYN STX 61 5C E2 C9 C7 D5 D6 D5 40 D9
D4 E3 F5 F5 ETX 45 A6 PAD PAD

*** RECEIVED DATA***
DLE AK1 PAD

*** TRANSMITTED DATA***
PAD SYN SYN SYN SYN EOT PAD PAD

*** END OF DIAGNOSTIC DUMP***

BIS-043 DIAGNOSTIC DUMP COMPLETE
======================================================
```

Table 4

4/BISYNC-80/3780 Configuration

Hardware Configuration

Before the Model II may be used to communicate with a REMOTE, Serial Channel A must be configured for synchronous communications. This must be done by a Radio Shack service technician.

Once this modification has been made, Serial Channel A can be used for synchronous communications using standard RS-232 cabling to the modem. To use Serial Channel A for asynchronous communications, pins 15, 17, and 24 of the asynchronous communications external DB-25 connector must be wired together. This is a special change required only because of the internal modifications made for bisynchronous communications.

In bisynchronous communications, characters (bytes) are "decoded" with the aid of an external clock (usually obtained from the modem). In asynchronous communications, characters (bytes) are "decoded" with the aid of an internal clock. Wiring pins 24, 15, and 17 together on the asynchronous DB-25 connector substitutes the internal clock signal for the external clock signal. This enables asynchronous communications to occur. Removal of this asynchronous DB-25 connector allows bisychronous cable connectors to be used.

Software Configuration

BISYNC-80/3780 is distributed on a TRSDOS diskette that contains four BISYNC-80/3780 files--one System file, one User file, and two User working files.

The System file is:

BIS3780--Mainline Executable Module.
This module and the Emulator User Configuration Executable Module are loaded into memory to execute BISYNC-80/3780.

The User file is:

- BIS3780C--User Configuration Module (Executable Module).

The User working files are:

- BIS3780C/MAC--The assembly source code of the User Configuration Executable Module, UCM.
- BIS3780C/LST--The assembly output listing of the User Configuration source code in a disk file form.

Note: The initial assembly and link of the User Configuration Module source code has been accomplished for you utilizing the Editor/Assembler (26-4702). The Editor/Assembler assembles the source code module BIS3780C/MAC producing, as output, BIS3780C/REL and the listing file BIS3780C/LST. Linking BIS3780C/REL and dumping the results produces BIS3780C.

All BISYNC-80/3780 parameters that you can change are contained in BIS3780C/MAC. To reconfigure the EMULATOR, you must have the Editor/Assembler and a knowledge of Assembly Language programming. You do not need to make any changes unless you want to change the parameters. (A listing of BIS3780C/MAC can be found in Appendix E.)

The User file parameters available for your modification are:

- Initial Device Mode (2770, 2780, 3741, 3780, or CPU)
- RJE Console Command identifiers and substitution strings
- ASCII/EBCDIC translation tables
- Filename extensions for ASCII/EBCDIC code translation

Table 5 lists all User Configuration Module (UCM) parameters, the names of the User Configuration Module statements that affect each parameter, and the possible parameter values. Most common changes can be made by changing simple equate statements or tables.

User Configuration Module Parameters

Configuration Parameter	UCM Statement Name	Possible Values	Supplied Distribution Disk Value
Device Mode	Device	M2770 M2780 M3780 M3741 MCPU	M3780
RJE Command Identifier	CMDTAB	Any character string	$(HASP) *(POWER)
Translate Tables	ASCEBC EBCASC	Any character string	IBM standard
Filename Extension Table For No-Translate Files	TSLTAB	Any character string	/REL /OBJ

Table 5

Device Mode Selection

On start-up, the EMULATOR uses M3780 (Device Mode 3780).

When the EMULATOR is initialized, it will be running in either 2770, 2780, 3741, 3780, or CPU-CPU mode. The UCM parameter DEVICE may be changed to cause this start-up mode to be any of the five modes.

- **M2770** specifies the EMULATOR will start up in 2770 mode.
- **M2780** specifies the EMULATOR will start up in 2780 mode.
- **M3741** specifies the EMULATOR will start up in 3741 mode.
- **M3780** specifies the EMULATOR will start up in 3780 mode.
- **MCPU** specifies the EMULATOR will start up in CPU-CPU mode.

Since a REMOTE may begin transmitting before you have a chance to enter a MODE command which changes the start-up mode to the proper value, we strongly recommend the initial mode setting be changed to match that of the REMOTE. (This mode setting must be accomplished using the Editor/Assembler and the procedures specified below.)

Example

To change the initial start-up mode to 2780 mode, change the BISY3780C/MAC statement named DEVICE from:

```
    DEVICE   EQU   M3780
```

to:

```
    DEVICE   EQU   M2780
```

RJE Command Identifiers

The character strings used by the EMULATOR to identify console commands as RJE commands are specified by the table at the label CMDTAB in the UCM. This table contains the string length, the value of the string, and the address of a substitution string.

During EMULATOR operation, a command entered on the Model II display will be interpreted as an RJE command if the leading characters of the command match an entry in CMDTAB. If a match is found, the substitution string pointed to by CMDTAB will be substituted for the matched string and the resulting command will be transmitted to the REMOTE.

The distribution diskette supplied table provides for the following:

```
        $ is translated to /*$
        * is translated to *b̸..b̸
```

Normal RJE Console operation requires that the REMOTE must process the command entered and immediately return a response to the EMULATOR for display on the Model II. If the REMOTE does not reply, the next data transmission from the REMOTE will be displayed on the Model II in error.

To provide for sending commands that have no response (such as SIGNON, SIGNOFF, and some POWER commands) the high-order bit (the 80H bit) of the substitution string length is used. If this bit in the UCM is on for any substitute string, no response to the command is expected from the REMOTE and the next transmission from the REMOTE will go to the assigned RECEIVE or PUNCH device or file.

Example

To create a CMDTAB entry to allow sending an OS SIGNON or SIGNOFF from the Model II where no reply is expected, and the command identifier is to be an exclamation point (!):

Make an entry in CMDTAB for the identifier:

```
        DB  1       ;identifier length
        DB  '!'     ;identifier
        DW  NORSPS  ;address of substitute string
```

Add the substitute string to the UCM (outside of CMDTAB!) with the 80H bit of the length turned on:

```
NORSPS  EQU $       ;string name
        DB  2+80H   ;length with 80H bit on
        DB  '/*'    ;substitute string
```

In operation, entering

```
!SIGNON REMOTE55
```

on the Model II will cause the string

/*SIGNON REMOTE55

to be transmitted to the REMOTE, without an expected response.

By expanding the substitute string in this example to include the entire signon record, it is possible to have a single keyboard key send the signon and/or signoff to the REMOTE.

ASCII/EBCDIC Translate Tables

The translate tables used to translate between ASCII and EBCDIC are at the UCM labels ASCEBC and EBCASC in the UCM.

The distribution diskette supplied tables conform to the IBM standard.

Filename Extension for No-Translate Files

The table at TSLTAB in the UCM defines the filename extensions of the files that are not to be translated between ASCII and EBCDIC by default.

The distribution diskette supplied UCM specifies that files with extensions of /OBJ and /LNK are not to be translated, by default.

Assembly, Link, and Dump Procedure

After changing your UCM, the UCM must be Assembled, Linked, and Dumped to an executable file. (See your Editor/Assembler manual for details on how to assemble and link the Module.) After this is done, type the following DUMP command to save the updated module:

```
    DUMP BIS3780C START=3000,END=3400,RORT=R <ENTER>
```

Reconfiguration Example

Since the EMULATOR is configured with 3780 mode at start-up, you will need to make the following change (using the Editor/Assembler) if you want 2780 mode at start-up.

Table 5 shows that the initial start-up device mode is controlled by the UCM statement named DEVICE. To change the start-up device mode from 3780 to 2780, the statement DEVICE EQU M3780 would need to be changed to DEVICE EQU M2780.

After you've edited the UCM, the UCM must be Assembled and Loaded.

To Assemble the Module, use the Editor/Assembler and type:

```
M80 BIS3780C,BIS3780C=BIS3780C <ENTER>
```

To Link the Module, after the assembly, type:

```
L80 BIS3780C <ENTER>
-E <ENTER>
```

After the Link, use the DUMP command to save the new Module:

```
DUMP BIS3780C START=3000,END=3400,RORT=R <ENTER>
```

Example

To Assemble and Load a Configuration Module, type:

```
M80 BIS3780C,BIS3780C=BIS3780C <ENTER>
L80 BIS3780C <ENTER>
-E <ENTER>
DUMP BIS3780C START=3000,END=3400,RORT=R <ENTER>
```

Verifying Installation and Configuration

Before running the EMULATOR's Analog Loopback Test (see EMULATOR Diagnostics), be sure that:

- All hardware modifications have been completed.
- All cables are properly connected.
- Any necessary changes to the UCM parameters have been made.
- The UCM has been Assembled, Linked, and Dumped.

Once you're sure all of these have been completed, run the Analog Loopback Test before executing BIS3780.

Appendix A/BISYNC-80/3780 Numbered Console Messages

BIS-001 INVALID COMMAND

The command entered was not a valid BISYNC-80 command. Re-enter a correct command.

BIS-002 TRANSMITTING

The EMULATOR has secured permission to send to the REMOTE and is transmitting the file(s) specified in a previous SEND command.

BIS-003 RECEIVING

The EMULATOR has granted the REMOTE permission to send, has opened the proper receive file (if needed) and is receiving data from the REMOTE.

BIS-004 TRANSMISSION COMPLETE

The EMULATOR has successfully completed the current transmission to the REMOTE.

BIS-005 MESSAGE RECEIVED COMPLETE

The EMULATOR has successfully received the current transmission from the REMOTE. The current receive file (if any) is closed.

BIS-006 CONSOLE BUFFER OVERFLOW--MESSAGE LOST

The EMULATOR was unable to display all messages generated for the Model II and has lost a message. This situation should occur only in high message volume situations (high numbers of errors) when the Model II cannot meet the demand for displaying messages. Correct the condition indicated by the high volume error message.

BIS-007 RECEIVE FILE DEFINITION REQUIRED

The REMOTE is attempting to send data to the EMULATOR's printer but no file or device has been defined for the data. The EMULATOR waits for a RECEIVE command. Enter a RECEIVE command to define a file or printer for the data.

BIS-008 PUNCH FILE DEFINITION REQUIRED

The REMOTE is attempting to send data to the EMULATOR's punch, but no file or device has been defined for the data. The EMULATOR waits for a PUNCH command. Enter a PUNCH command to define a file or printer for the data.

BIS-009 DUPLICATE FILE--DELETE? (Y or N)
FILENAME filespec

filespec has been specified for data from the REMOTE, but the file already exists. Type <Y> <ENTER> to delete the existing file and create a new file of the same name for data from the REMOTE. Type <N> <ENTER> if the existing file is not to be deleted.The EMULATOR will prompt for a new file or device name.

BIS-010 REMOTE ABORT

The current operation in progress has been aborted by the REMOTE. If the operation was a receive from the REMOTE, the current RECEIVE or PUNCH file (if any) is closed. If Receive Chaining is not in effect, the RECEIVE or PUNCH command is satisfied. If the operation was a transmit to the REMOTE, it was only partially completed. The EMULATOR returns to the idle state.

BIS-011 REMOTE DISCONNECTED

A disconnect sequence has been received from the REMOTE. If further operations with the REMOTE are desired, a new data connection must be established.

BIS-012 DIAGNOSE FILE OPEN ERROR

The file specified in a DIAGNOSE command could not be opened. Be sure a valid filename is specified and re-enter the DIAGNOSE command.

BIS-013 SYSTEM IS IDLE - NO ABORT REQUIRED

No operation was in progress when an ABORT command was entered. No abort is possible.

BIS-014 TRANSMISSION ABORTED BY OPERATOR

The current transmission to the REMOTE has been ended by the ABORT command. The EMULATOR returns to the idle state.

BIS-015 RECEPTION ABORTED BY OPERATOR

The current data reception from the REMOTE has been ended by the ABORT command. The current RECEIVE or PUNCH file (if any) is closed, and (if Receive Chaining is not in effect) the current RECEIVE or PUNCH command is satisfied. The EMULATOR returns to the idle state.

```
BIS-016 SYSTEM STATUS
SYSTEM IS  activity , IN  device  DEVICE MODE
TRANSPARENCY  is or is not  IN EFFECT  transparency mode
RECEIVE CHAINING  is or is not  IN EFFECT  chaining mode
CURRENT FILES ASSIGNED:
RECEIVE FILE    -   filespec
PUNCH FILE      -   filespec
TRANSMISSION LINE STATISTICS
SENT  RECEIVED
TOTAL BLOCKS:          nnnn            nnnn
TOTAL NAKS:            nnnn            nnnn
TOTAL ENQS:            nnnn            nnnn
CURRENT BLOCKS:        nnnn            nnnn
CURRENT NAKS:          nnnn            nnnn
CURRENT ENQS:          nnnn            nnnn
***END OF STATISTICS DISPLAY***
```

This message lists the current EMULATOR operation and modes and shows block and error statistics for the line.

BIS-017 INVALID FILE NAME SPECIFIED

A filename in the previous command was incorrectly specified. Re-enter the command using correct syntax or filename.

BIS-018 LOOPBACK TEST--
ENTER ANY CHARACTER TO TERMINATE TEST

After ALTEST has been entered, the EMULATOR is running the Analog Loopback Test. The Test will continue until any Model II keyboard character is entered or until an error occurs. If no error is reported, enter a keyboard character (or simply press <ENTER>) to terminate the Test.

BIS-019 SYSTEM NOT IDLE--ALTEST IGNORED

An ALTEST command has been entered, but the EMULATOR is not in the idle state. The ALTEST command is ignored. Allow the current operation to complete, then re-enter the ALTEST command.

BIS-020 NO DATA PATH ESTABLISHED--TEST FAILED

The Analog Loopback Test did not receive back the data transmitted. The EMULATOR program is not operational. See EMULATOR Diagnostics, Table 3 for a list of the possible failure causes.

BIS-021 DATA PATH ERROR--TEST FAILED
DATA EXPECTED - nn DATA RECEIVED - nn

The data received by the Analog Loopback Test did not match the data sent. nn in the message text indicates the hexadecimal values of the byte sent and the byte received. The EMULATOR is not operational. See EMULATOR Diagnostics, Table 3 for a list of the possible failure causes.

BIS-022 LOOPBACK TEST COMPLETED WITHOUT ERROR

The Analog Loopback Test has successfully completed. The EMULATOR is operational. Take the modem out of Analog Loopback.

BIS-023 INVALID CHANNEL STATUS

The serial channel status obtained by the EMULATOR is invalid. The EMULATOR is not operational. See EMULATOR Diagnostics, Table 3 for a list of the possible failure causes.

BIS-024 TRANSMITTER TIMEOUT

The Serial Channel transmitter being used by the EMULATOR would not accept a character for 50 ms. The EMULATOR is not operational. See EMULATOR Diagnostics, Table 3 for a list of the possible failure causes.

BIS-025 I/O ERROR HAS OCCURRED
FILENAME - filespec
STATUS nn

An unrecoverable file I/O error has occurred. The current operation in progress will be aborted. The TRSDOS status returned after the I/O request is represented in the message text by nn, where nn is a hexadecimal number. Convert this number to decimal to determine the TRSDOS error message number.

BIS-026 NO RESPONSE TO LINE BID

The EMULATOR has not received a response to 15 consecutive line bids. The REMOTE is not connected or is not prepared to receive data. Connect to the REMOTE or ask a person at the REMOTE to prepare the REMOTE for communications; then re-enter the SEND command.

BIS-027 LINE BID REFUSED

The REMOTE has refused the EMULATOR's line bid. The REMOTE is not prepared to receive data. Contact a person at the REMOTE and ask to have the REMOTE prepared for data transfer.

BIS-028 LOST ACK SYNCHRONIZATION--ABORTED

The REMOTE has sent the improper alternating acknowledgment in response to a data block. This is a protocol problem of the REMOTE. The current transmission is aborted. Try the operation again by re-entering the SEND command. If the problem recurs, enter the DIAGNOSE command, save the results, and contact a support person at the REMOTE.

BIS-029 15 RETRIES UNSUCCESSFUL--ABORTED

Fifteen successive attempts to transmit a data block were not successful. The transmission is aborted. This failure is probably due to a faulty data connection. Re-establish the data connection using another line, if possible. If the error persists, run the Analog Loopback Test and, if this Test is successful, run end-to-end modem tests as described in your modem documentation.

BIS-030 MODE HAS BEEN SET

The Mode specified by the previous mode command has been set.

BIS-031 DIAGNOSTIC DUMP

This message precedes the diagnostic dump produced by the DIAGNOSE command and identifies the data as a diagnostic dump.

BIS-032 CONSOLE NOT ASSIGNED TO FILE

A <CTRL><E> was entered on the Model II, but the EMULATOR command input was not assigned to a file. The <CTRL><E> is discarded.

BIS-033 FILE NOT FOUND
FILENAME - filespec

The file shown in the message (specified in a previous SEND command) was not found. Re-enter the SEND command, using the correct filename.

BIS-034 CONSOLE HAS BEEN ASSIGNED

In response to a previous CONSOLE command, the EMULATOR command input has been assigned to the file specified by the command.

BIS-035 EXIT COMMAND READ OR COMMAND FILE EOF
SYSTEM CONSOLE ACTIVE

An end-of-file or EXIT command has been encountered in a command file. The EMULATOR command input has been re-assigned to the Model II keyboard.

BIS-036 COMMAND REJECTED-COMMAND QUEUE FULL

Two SEND, RECEIVE, or PUNCH commands were queued for execution when a third SEND, RECEIVE, or PUNCH command was entered. The command is rejected. Wait until one of the outstanding commands have been executed, then re-enter the command.

BIS-037 INVALID CHARACTER IN TRANSMIT FILE-ABORTED

A character that is also a line control character has been detected in a file being transmitted non-transparently. The transmit operation has been aborted. This error usually occurs because of an attempt to transmit a binary file or an untranslated ASCII file in Non-Transparent Mode. Be sure the translate and transparency attributes are correctly specified; then try the operation again.

BIS-038 TRANSMIT FILE RECORD TOO LONG-TRUNCATED

A record longer than 80 characters was found in a file being transmitted with ASCII/EBCDIC translation. The record was truncated before transmission. Be sure all records in files being transmitted with translation are less than 81 characters in length.

BIS-039 TIMEOUT HAS OCCURRED - ABORTED

The EMULATOR has not received a response to 15 consecutive ENQs requesting acknowledgement of previously transmitted data block. Check that the communications link and REMOTE are functioning correctly.

BIS-040 EXITED BISYNC-80/3780

Control has returned to TRSDOS.

BIS-041 SYSTEM CONSOLE ACTIVE

In response to a <CTRL><E>, control has passed from a command file to a Model II keyboard. You may enter EMULATOR commands at the Model II keyboard.

BIS-042 COMMAND SEQUENCE FILE ACTIVE

In response to a <CTRL><E> entered at the Model II keyboard, the EMULATOR has resumed processing commands from the previously assigned command file.

BIS-043 DIAGNOSTIC DUMP COMPLETE

The DIAGNOSE command has completed execution.

BIS-044 SYSTEM NOT IDLE - EXIT IGNORED

An EXIT command has been entered but the system is not idle. Wait for the current operation to complete and re-enter the EXIT command. The system may be forced back to the idle state by using the ABORT command.

BIS-045 UNABLE TO LOAD BIS3780C

The BISYNC-80 initialization routine was not able to load the BIS3780C User Configuration Module. Be sure the User Configuration Module has been configured, assembled, linked, dumped, and resides on the same diskette as the BIS3780 file.

BIS-046 DISK IS FULL - CHANGE REQUIRED

The diskette, which is to contain the file being received, is full. Enter a new RECEIVE or PUNCH command, specifying a file on a different disk drive or change diskette and enter a RECEIVE or PUNCH command for the new diskette.

BIS-047 RVI RECEIVED - TRANSMISSION SUSPENDED

The REMOTE has sent an RVI (Reverse Interrupt) sequence in response to a block. This probably indicates that the REMOTE has high priority messages to be sent to the EMULATOR. The current SEND operation is terminated.

BIS-048 CHANNEL A NOT DISABLED

The EMULATOR has found Serial Channel A initialized. Control was returned to TRSDOS. Use the **SETCOM A=OFF** command to free Serial Channel A and restart the EMULATOR.

BIS-049 PRINTER ERROR - STATUS nn
REPLY 'Y' TO CONTINUE OR 'N' TO ABORT

A printer error has occurred and nn is the TRSDOS status returned from the printer operation. nn is a hexadecimal number. The printer may be out of paper (status 2B), disabled (status 2A), not turned on or broken (status 2C), or not available (status 2C). Correct the fault, if possible, and type <Y> <ENTER> to continue. Type <N> <ENTER> to end the current print operation if the error cannot be corrected.

BIS-050 INVALID DATA FOR MODE - RECEPTION ABORTED

The format of the data received by the EMULATOR is not correct for the current Device Mode (2770,2780,3741,3780,or CPU). The operation in progress is aborted. Be sure the EMULATOR's device mode matches the mode expected by the REMOTE.

BIS-051 DEQUEUE FOR device FILE ACCEPTED

The previous DEQUEUE command has been accepted. <u>device</u> is RECEIVE if the RECEIVE device is to be dequeued <u>or</u> PUNCH if the PUNCH device is to be dequeued. The dequeue operation is performed immediately if the EMULATOR is not receiving; otherwise, the dequeue operation is performed immediately after the reception.

BIS-052 DEQUEUE FOR <u>device</u> FILE OUTSTANDING

A DEQUEUE command could not be processed because a previous, unprocessed DEQUEUE command is still outstanding. Wait until the current receive operation is complete; then re-enter the command.

Appendix B/Forms Control Sequence Decoding

Printer forms control escape sequences received in translated transmissions from the REMOTE will be translated into forms control sequences. See Table 6.

Escape Sequence	**Translated To:**	**Printer**
Action		
ESC /	(CR)	Single space
ESC S	(CR) (CR)	Double space
ESC T	(CR) (CR) (CR)	Triple space
ESC A	(CR) (FF)	Form feed
ESC M		Suppress space
ESC HT		Save horizontal format record
All others	(CR)	Single space

Table 6

An ESC HT sequence is processed by the EMULATOR as soon as it is received. Other ESC sequences are processed after receiving an NL or IRS character (3780 mode), an IUS character (2780 mode), or an ETX or ETB character (other modes) in keeping with the IBM device protocol.

Other EBCDIC forms control characters received from the line are also translated. See Table 7.

ASCII		**Printer**
EBCDIC Character	**Translation**	**Action**
(FF)	(CR) (FF)	Form feed
(NL)	(LF)	Single space
(VT)	(CR) (FF)	Form feed
(LF)	(CR)	Single space
(IRS) or (IUS) with no outstanding ESC	(CR)	Single space

Table 7

Forms control processing is not performed for non-translated data.

Appendix C/Binary Synchronous Communications (BSC) Protocol Summary

BSC Control Characters

The BSC protocol provides a set of rules for synchronous transmission of data. All data is transmitted as a string of binary digits. The transmitter and receiver establish and maintain synchronization through recognition of a specific bit pattern (called the sync pattern) that is transmitted at the beginning of each transmission block.

Control characters are inserted into each data block by the transmitter and deleted from the block by the receiver. These characters are used to delimit records within a block, indicate the start and end of a block, and control such functions as blank compression. Other control characters are used to insure the orderly and accurate transfer of data between transmitter and receiver.

Table 8 lists the hex values of the BSC Control Characters used by the EMULATOR.

Table 9 lists the Control Characters, their meanings, and the level of EMULATOR use.

BSC Control Character Values

Control Character	EBCDIC Hex Value
ACKO	10H, 70H - two bytes
ACK1	10H, 61H - two bytes
DLE	10H
ENQ	2DH
EOT	37H
ESC	27H
ETB	26H
ETX	03H
IRS	1EH
IUS	1FH
NAK	3DH
PAD	FFH
RVI	10H, 7CH - two bytes
SOH	01H
STX	02H
SYN	32H
WACK	10H, 6BH - two bytes

Table 8

BSC Control Characters Used by the EMULATOR

Control Character	Meaning and Use
ACK0	Positive response to line bid. Positive response to data block, alternating with ACK1 to insure no blocks are missed.
ACK1	Positive response to data block, alternating with ACK0 to insure no blocks are missed.
DLE	Provides two-character control sequences (ACK0, ACK1, RVI, WACK). Also used to indicate control characters in transparent mode.
ENQ	Enquiry. Line bid, or request response to last block sent if response is not received.
EOT	End-of-Transmission. Releases the line for a new line bid.
ESC	Escape. Precedes and identifies forms and format control sequences.
ETB	End-of-Text Block. Signifies the end of the current block and indicates that the block check sequence follows immediately. More data blocks follow after reply from receiver.
ETX	End-of-Text. Logical end-of-file. Indicates the end of the current block. Block check sequence follows. No more data blocks follow.
IRS	Delimits 3780 records.
IUS	Delimits 2780 records. Causes error checking without line turn-around.
NAK	Negative reply to line bid or data block. Transmitter must re-transmit.
PAD	Preceeds and follows all data transmitted. Allows for line turn-around.

RVI	Reverse Interrupt. Not transmitted by the EMULATOR but acted upon if received. Causes suspension of the current transmission and relinquishes control of the communication line to the REMOTE for a high-priority message.
SOH	Start-of-Header. Not transmitted by the EMULATOR. Treated as STX if received from a REMOTE.
STX	Start-of-Text. Marks the beginning of a data block. Starts the block error check accumulation.
SYN	Character Synchronization. Precedes all message and control blocks and establishes character phase. Two consecutive SYNs are required for proper sync. The EMULATOR transmits four consecutive SYNs to the REMOTE.
WACK	Wait Acknowledgment. Acknowledges the previous line bid or data block and requests that the transmitter pause before sending more data.

Table 9

BSC Data Transfer

BSC data transfer operations are initiated by a process known as a "line bid." The station initiating data transfer requests permission to send by transmitting an ENQ character to the REMOTE. If the REMOTE is prepared to receive data, it responds positively, sending back an ACKO reply. If the REMOTE is not prepared to receive data, it will respond with a negative reply, or NAK.

Once permission to transmit has been secured, the transmitting station formats and transmits a data block. This data block contains line control characters, data, and a cyclic redundancy accumulation used for error detection. If the receiving station receives the block and the cyclic redundancy check calculated by the receiver matches the one sent by the transmitter, the data is accepted by the receiver and the receiver replies with an alternating positive acknowledgement. An ACKO is sent in response to the line bid, an ACK1 is sent in response to the first data block, an ACKO is sent in response to the second data block, and so forth, with the ACKs alternating between ACKO and ACK1.

If the receiver's cyclic redundancy check calculation does not match the transmitted calculation, a line error has occurred and the receiver responds with a NAK. The transmitter re-transmits the block, attempting to secure a positive reply.

If the transmitter does not receive a reply to a block, either the receiver did not receive the block or the receiver's reply was not received by the transmitter. The transmitter transmits an ENQ character, requesting the receiver to resend the last block check reply. The reply received will be either:

- NAK (indicating the receiver received the block with an error).
- The proper alternating ACK (indicating the receiver received the block without error).
- The improper alternating ACK (indicating the receiver did not receive the block).

The transmitter must then resend the current block or send the next block, depending on the reply.

End-of-transmission, either normal or abnormal, is indicated by the transmitter or receiver sending an EOT character.

Table 10 shows sample protocol used to transfer data between stations. This figure is representative of the protocol used, but does not show all possible transmitter and receiver actions. For more information, you are encouraged to consult the IBM publications on BSC protocol listed earlier in this manual.

Typical Line Protocol

Normal Transmission:

```
    TRANSMITTER:    ENQ       DATA1       DATA2          EOT
                       \      /    \      /     \         /
    RECEIVER:           ACK0        ACK1          ACK0
```

Block Rejected-CRC Error

```
    TRANSMITTER:    ENQ       DATA1        DATA1   DATA2    EOT
                       \      /    \       /    \   /    \   /
    RECEIVER:           ACK0        NAK         ACK1     ACK0
```

No Response from Receiver

```
    TRANSMITTER:    ENQ       DATA1   ENQ        DATA2    EOT
                       \      /          \       /    \    /
    RECEIVER:           ACK0              ACK1        ACK0
```

Table 10

Appendix D/Cable Requirements

A standard RS-232-C 11-pin cable is required for connection between the TRS-80 Model II system and the modem. This cable must be equipped with male DB-25 type connectors on both ends, with the following pins connected:

DB-25 Pin Designation

1	AA - Protective Ground
2	BA - TXD Transmitted Data
3	BB - RXD Received Data
4	CA - RTS Request to Send
5	CB - CTS Clear to Send
6	CC - DSR Data Set Ready
7	AB - Signal Ground
8	CF - DCD Carrier Detect
15	DB - TXC Transmit Clock
17	DD - RXC Receive Clock
20	CD - DTR Data Terminal Ready

Appendix E/User Configuration Module Listing

```
00010   ;
00020   ;
00030   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
00040   ;
00050   ;  (C) COPYRIGHT 1981 MICRO-INTEGRATION, INC.
00060   ;      LICENSED TO TANDY CORPORATION
00070   ;
00080   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
00090   ;
00100   ;
00200   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
00300   ;                                                               ;
00400   ;                          BISYNC80/3780                        ;
00500   ;                    USER CONFIGURATION MODULE                  ;
00600   ;                                                               ;
00700   ;                                                               ;
00800   ;       THIS MODULE MUST BE CHANGED IF THE LINK'S INITIAL       ;
00900   ;       I/O PORT ADDRESSES OF RUN TIME PARAMETERS ARE TO BE     ;
01000   ;       CHANGED.                                                ;
01100   ;                                                               ;
01200   ;                                                               ;
01300   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
01400   ;
01500   TRUE    EQU     1
01600   FALSE   EQU     0
01700   M2780   EQU     0
01800   M3780   EQU     2
01900   M2770   EQU     4
02000   M3741   EQU     6
02100   MCPU    EQU     8
02200   ALLIO   EQU     1
02300   IN8251  EQU     2
02400   Z80SIO  EQU     3
02500   CPM     EQU     5
02600           ASEG
02700   BASEAD  EQU     03000H          ;BASE ADDRESS FOR THIS MODULE
02800           ORG     BASEAD          ;ORG TO MODULE START ADDRESS
02900   .8080
03000           JMP     RTMODE          ;RETURN DEVICE MODE
03100           JMP     RTCMTB          ;RETURN COMMAND TABLE ADDRESS
03200           JMP     RTXLAT          ;RETURN TRANSLATE TABLE ADDRESS
03300           JMP     RTFLEX          ;RETURN NOTRANSLATE FILE EXTENSIONS
03400           JMP     RTRTSA          ;RETURN REQUEST TO SEND ASSERT MODE
03500   ;
03600   ;       THE FOLLOWING EQUATE STATEMENTS MAY BE CHANGED
03700   ;       TO RECONFIGURE THE LINK'S FACTORY SUPPLIED PARAMETERS FOR --
03800   ;
03900   ;               -       2770, 2780, 3741, 3741, OR CPU STARTUP MODE
04000   ;               -       CONTINUOUS REQUEST TO SEND ASSERTION
04100   ;               -       RJE COMMAND IDENTIFIER
```

```
04200   ;               -       TRANSLATE TABLES
04300   ;               -       FILENAME EXTENSIONS
04400   ;
04500   DEVICE  EQU     M3780           ;CHANGE THIS STATEMENT TO 'M2780'
04600   ;                               ;TO CHANGE THE INITIAL STARTUP
04700   ;                               ;DEVICE MODE TO 2780 MODE, TO
04800   ;                               ;'M2770' TO CHANGE THE INITIAL
04900   ;                               ;DEVICE MODE TO 2770 MODE, TO
05000   ;                               ;'M3741' TO CHANGE THE INITIAL
05100   ;                               ;DEVICE MODE TO 3741 MODE, OR TO
05200   ;                               ;'MCPU' TO CHANGE THE INITIAL
05300   ;                               ;STARTUP DEVICE MODE TO CPU MODE.
05400   ;                               ;LEAVE THIS STATEMENT AS-IS TO HAVE
05500   ;                               ;THE INITIAL STARTUP MODE REMAIN
05600   ;                               ;3780 MODE.
05700   ;
05800   RTSDEF  EQU     FALSE           ;CHANGE THIS STATEMENT TO 'TRUE'
05900   ;                               ;TO CAUSE CONTINUOUS ASSERTION OF
06000   ;                               ;THE MODEM CONTROL LINE 'REQUEST
06100   ;                               ;TO SEND' (RTS).  THIS ASSERTION
06200   ;                               ;IS VALID ON DEDICATED,
06300   ;                               ;FULL-DUPLEX COMMUNICATIONS
06400   ;                               ;FACILITIES ONLY, AND CONNOT BE
06500   ;                               ;ENABLED FOR HALF-DUPLEX FACILITIES.
06600   ;
06700   ;
06800   ;
06900   ;
07000   ;       THE FOLLOWING TABLE DEFINES THE HASP AND POWER CONSOLE
07100   ;       COMMAND IDENTIFIERS AND THE STRINGS TO BE SUBSTITUTED
07200   ;       FOR THE IDENTIFIER WHEN TRANSMITTING A CONSOLE COMMAND.
07300   ;       THE IDENTIFIERS AND/OR THE STRINGS MAY BE MODIFIED TO CAUSE
07400   ;       TRANSMISSION OF ANY STRING FOR ANY IDENTIFIER.
07500   ;
07600   ;       TABLE FORMAT :
07700   ;               1 BYTE IDENTIFIER LENGTH
07800   ;               N BYTES OF IDENTIFIER
07900   ;               2 BYTE ADDRESS OF SUBSTITUTE STRING
08000   ;
08100   ;               END OF TABLE DELIMITED BY A LENGTH BYTE
08200   ;               WITH A VALUE OF ZERO.
08300   ;
08400   ;       SUBSTITUTE STRING FORMAT:
08500   ;               1 BYTE STRING LENGTH
08600   ;               N BYTES OF STRING
08700   ;
08800   ;
08900   ;
09000   ;
09100   CMDTAB  EQU     $
```

```
09200           DB      1                       ;IDENTIFIER LENGTH (MAX 255)
09300           DB      '$'                     ;IDENTIFIER
09400           DW      HASPCM                  ;ADDRESS OF STRING TO SUBSTITUTE
09500   ;                                       ;FOR IDENTIFIER
09600           DB      1                       ;IDENTIFIER LENGTH
09700           DB      '*'                     ;IDENTIFIER
09800           DW      POWRCM                  ;ADDRESS OF STRING TO SUBSTITUTE
09900   ;                                       ;FOR IDENTIFIER
10000   ;
10100           DB      0                       ;END OF TABLE - DO NOT REMOVE THIS
10200   ;                                       ;STATEMENT
10300   ;
10400   HASPCM: DB      3                       ;SUBSTITUTE STRING LINGTH
10500           DB      '/*$'                   ;SUBSTITUTE STRING
10600   ;
10700   POWRCM: DB      5                       ;SUBSTITUTE STRING LENGTH
10800           DB      '* .. '                 ;SUBSTITETE STRING
10900   ;
11000   ;
11100   ;
11200   ;
11300   ;       THE FOLLOWING TWO TABLES ARE THE ASCII TO EBCDIC
11400   ;       AND EBCDIC TO ASCII TRANSLATE TABLES USED BY THE
11500   ;       LINK IN CODE CONVERSION.  THESE TABLES MAY BE
11600   ;       MODIFIED TO ALLOW TRANSLATION OF ANY CHARACTER TO
11700   ;       ANY OTHER CHARACTER, SO LONG AS LINE CONTROL
11800   ;       CHARACTERS ARE NOT TRANSMITTED IN NON-TRANSPARENT
11900   ;       MODE.
12000   ;
12100   ;
12200   ASCEBC  EQU     $                       ;ASCII TO EBCDIC TRANSLATE TABLE
12300   ;
12400           DB      000H,001H,002H,003H,037H,02DH,02EH,02FH  ;0-7
12500           DB      016H,005H,025H,00BH,00CH,00DH,00EH,00FH  ;8-0FH
12600           DB      010H,011H,012H,013H,03CH,03DH,032H,026H  ;10H-17H
12700           DB      018H,019H,03FH,027H,01CH,01DH,01EH,01FH  ;18H-1FH
12800           DB      040H,05AH,07FH,07BH,05BH,06CH,050H,07DH  ;20H-27H
12900           DB      04DH,05DH,05CH,04EH,06BH,060H,04BH,061H  ;28H-2FH
13000           DB      0F0H,0F1H,0F2H,0F3H,0F4H,0F5H,0F6H,0F7H  ;30H-37H
13100           DB      0F8H,0F9H,07AH,05EH,04CH,07EH,06EH,06FH  ;38H-3FH
13200           DB      07CH,0C1H,0C2H,0C3H,0C4H,0C5H,0C6H,0C7H  ;40H-47H
13300           DB      0C8H,0C9H,0D1H,0D2H,0D3H,0D4H,0D5H,0D6H  ;48H-4FH
13400           DB      0D7H,0D8H,0D9H,0E2H,0E3H,0E4H,0E5H,0E6H  ;50H-57H
13500           DB      0E7H,0E8H,0E9H,0ADH,0E0H,0BDH,05FH,06DH  ;58H-5FH
13600           DB      079H,081H,082H,083H,084H,085H,086H,087H  ;60H-67H
13700           DB      088H,089H,091H,092H,093H,094H,095H,096H  ;68H-6FH
13800           DB      097H,098H,099H,0A2H,0A3H,0A4H,0A5H,0A6H  ;70H-77H
13900           DB      0A7H,0A8H,0A9H,0BBH,06AH,09BH,0A1H,007H  ;78H-7FH
14000           DB      000H,001H,002H,003H,037H,02DH,02EH,02FH  ;80H-87H
14100           DB      016H,005H,025H,00BH,00CH,00DH,00EH,00FH  ;88H-8FH
```

```
14200           DB      010H,011H,012H,013H,03CH,03DH,032H,026H  ;90H-97H
14300           DB      018H,019H,03FH,027H,01CH,01DH,01EH,01FH  ;98H-9FH
14400           DB      040H,05AH,07FH,07BH,05BH,06CH,050H,07DH  ;A0H-A7H
14500           DB      04DH,05DH,05CH,04EH,06BH,060H,04BH,061H  ;A8H-AFH
14600           DB      0F0H,0F1H,0F2H,0F3H,0F4H,0F5H,0F6H,0F7H  ;B0H-B7H
14700           DB      0F8H,0F9H,07AH,05EH,04CH,07EH,06EH,06FH  ;B8H-BFH
14800           DB      07CH,0C1H,0C2H,0C3H,0C4H,0C5H,0C6H,0C7H  ;C0H-C7H
14900           DB      0C8H,0C9H,0D1H,0D2H,0D3H,0D4H,0D5H,0D6H  ;C8H-CFH
15000           DB      0D7H,0D8H,0D9H,0E2H,0E3H,0E4H,0E5H,0E6H  ;D0H-D7H
15100           DB      0E7H,0E8H,0E9H,0ADH,0E0H,0BDH,05FH,06DH  ;D8H-DFH
15200           DB      079H,081H,082H,083H,084H,085H,086H,087H  ;E0H-E7H
15300           DB      088H,089H,091H,092H,093H,094H,095H,096H  ;E8H-EFH
15400           DB      097H,098H,099H,0A2H,0A3H,0A4H,0A5H,0A6H  ;F0H-F7H
15500           DB      0A7H,0A8H,0A9H,0BBH,06AH,09BH,0A1H,007H  ;F8H-FFH
15600   ;
15700   EBCASC  EQU     $                       ;EBCDIC TO ASCII TRANSLATE TABLE
15800   ;
15900           DB      000H,001H,002H,003H,000H,009H,000H,07FH  ;00H-07H
16000           DB      000H,000H,000H,00BH,00CH,00DH,00EH,00FH  ;08H-0FH
16100           DB      010H,011H,012H,013H,000H,00AH,008H,000H  ;10H-17H
16200           DB      018H,019H,000H,000H,01CH,01DH,01EH,01FH  ;18H-1FH
16300           DB      000H,000H,01CH,000H,000H,00AH,017H,01BH  ;20H-27H
16400           DB      000H,000H,000H,000H,000H,005H,006H,007H  ;28H-2FH
16500           DB      000H,000H,016H,000H,000H,01EH,000H,004H  ;30H-37H
16600           DB      000H,000H,000H,000H,014H,015H,000H,01AH  ;38H-3FH
16700           DB      020H,000H,000H,000H,000H,000H,000H,000H  ;40H-47H
16800           DB      000H,000H,000H,02EH,03CH,028H,02BH,021H  ;48H-4FH
16900           DB      026H,000H,000H,000H,000H,000H,000H,000H  ;50H-57H
17000           DB      000H,000H,021H,024H,02AH,029H,03BH,05EH  ;58H-5FH
17100           DB      02DH,02FH,000H,000H,000H,000H,000H,000H  ;60H-67H
17200           DB      000H,000H,07CH,02CH,025H,05FH,03EH,03FH  ;68H-6FH
17300           DB      000H,000H,000H,000H,000H,000H,000H,000H  ;70H-77H
17400           DB      000H,060H,03AH,023H,040H,027H,03DH,022H  ;78H-7FH
17500           DB      000H,061H,062H,063H,064H,065H,066H,067H  ;80H-87H
17600           DB      068H,069H,000H,07BH,000H,000H,000H,000H  ;88H-8FH
17700           DB      000H,06AH,06BH,06CH,06DH,06EH,06FH,070H  ;90H-97H
17800           DB      071H,072H,000H,07DH,000H,000H,000H,000H  ;98H-9FH
17900           DB      000H,07EH,073H,074H,075H,076H,077H,078H  ;A0H-A7H
18000           DB      079H,07AH,000H,000H,000H,05BH,000H,000H  ;A8H-AFH
18100           DB      030H,031H,032H,033H,034H,035H,036H,037H  ;B0H-B7H
18200           DB      038H,039H,000H,000H,000H,05DH,000H,05FH  ;B8H-BFH
18300           DB      000H,041H,042H,043H,044H,045H,046H,047H  ;C0H-C7H
18400           DB      048H,049H,000H,000H,000H,028H,02BH,02BH  ;C8H-CFH
18500           DB      000H,04AH,04BH,04CH,04DH,04EH,04FH,050H  ;D0H-D7H
18600           DB      051H,052H,000H,07DH,000H,029H,000H,000H  ;D8H-DFH
18700           DB      05CH,000H,053H,054H,055H,056H,057H,058H  ;E0H-E7H
18800           DB      059H,05AH,000H,000H,000H,000H,000H,000H  ;E8H-EFH
18900           DB      030H,031H,032H,033H,034H,035H,036H,037H  ;F0H-F7H
19000           DB      038H,039H,000H,000H,000H,000H,000H,000H  ;F8H-FFH
19100   ;
```

```
19200   ;
19300   ;       THE FOLLOWING TABLE DEFINES THE FILENAME EXTENSIONS FOR
19400   ;       FILES THAT ARE NOT TO BE TRANSLATED TO AND FROM EBCDIC.
19500   ;       THE TABLE HAS THE SAME FORMAT AS THE RJE CONSOLE IDENTIFIER
19600   ;       TABLE SHOWN ABOVE.  THE TABLE MUST HAVE AT LEAST ONE ENTRY.
19700   ;
19800   ;
19900   TSLTAB: DB      3               ;ENTRY LENGTH
20000           DB      'REL'           ;REL FILES ARE NOT TRANSLATED
20100           DB      0               ;REQUIRED
20200           DB      3               ;ENTRY LENGTH
20300           DB      'OBJ'           ;OBJ FILES ARE NOT TRANSLATED
20400           DW      0               ;REQUIRED
20500   ;                               ;TRANSLATED.  NOTE THAT THE STRING
20600   ;
20700           DB      0               ;END OF TABLE - REQUIRED
20800   ;
20900   ;
21000   ;       THE FOLLOWING STATEMENTS MUST NOT BE MODIFIED BY THE USER.
21100   ;
21200   ;
21300   ;
21400   ;       THE FOLLOWING ROUTINES MUST NOT BE USER MODIFIED.
21500   RTMODE: MVI     A,DEVICE        ;GET DEVICE MODE
21600           RET                     ;BACK TO CALLER
21700   RTCMTB: LXI     H,CMDTAB        ;GET CONSOLE COMMAND TABLE ADDRESS
21800           RET                     ;BACK TO CALLER
21900   RTXLAT: LXI     H,ASCEBC        ;GET TRANSLATE TABLE ADDRESS
22000           RET                     ;BACK TO CALLER
22100   RTFLEX: LXI     H,TSLTAB        ;GET NOTRANSLATE FILE EXTENSION TABLE
22200           RET                     ;BACK TO CALLER
22300   RTRTSA: MVI     A,RTSDEF        ;GET RTS ASSERT MODE BYTE
22400           RET                     ;BACK TO CALLER
22500   ;
22600   ;
22700           END     3000H           ;END - ORG AT 3000H
```

Appendix F/EMULATOR Command Summary

Command Syntax	Use
ABORT	Ends (aborts) the current EMULATOR function and returns the EMULATOR to the Idle state.
ALTEST	Directs the EMULATOR to perform an Analog Loopback Test.
CONSOLE filespec	Assigns the EMULATOR command input device to a file.
DEQUEUE remove	Deletes (DEQUEUEs) the EMULATOR's current RECEIVE or PUNCH file to allow reassignment of the RECEIVE or PUNCH device.
DIAGNOSE filespec {PRT}	Dumps a history of recent communications line activity to the Model II display, printer, or to a disk file.
EXIT	Ends EMULATOR execution.
MODE mode trans chain	Sets the Device Mode to 2770, 2780, 3741, 3780, or CPU Mode; the Transmission Mode for Transparent (TSP) or Non-Transparent (NTSP) data transmission; the Receive Chaining Mode for Receive Chaining (RC) or No-Receive Chaining (NRC).
PUNCH filespec {option} {PRT}	Specifies the Device to be used or the name of the file which will contain punch-destined data (from the REMOTE).

RECEIVE filespec {option} {PRT}
Specifies the Device to be used to send data to the printer or the name of the file which will contain data to be sent to the printer (from the REMOTE).

SEND filespec trans
Transmits specified file or files to the REMOTE.

STAT
Creates a display on the Model II of the current EMULATOR status and error accumulators.

Index

IMPORTANT NOTICE

ALL RADIO SHACK COMPUTER PROGRAMS ARE LICENSED ON AN "AS IS" BASIS WITHOUT WARRANTY.

Radio Shack shall have no liability or responsibility to customer or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by computer equipment or programs sold by Radio Shack, including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such computer or computer programs.

NOTE: Good data processing procedure dictates that the user test the program, run and test sample sets of data, and run the system in parallel with the system previously in use for a period of time adequate to insure that results of operation of the computer or program are satisfactory.

RADIO SHACK SOFTWARE LICENSE

Notwithstanding the terms and conditions of the Radio Shack Computer Sales Agreement signed by Customer, the following license terms will be applicable only to this BISYNC-80/3780 (Radio Shack Catalog No. 26-4716) Package (software and manual):

A. Radio Shack grants to CUSTOMER a non-exclusive, paid up license to use on any of CUSTOMER'S computers the Radio Shack computer software BISYNC-80/3780 Package. Title to the media on which the software is recorded is transfered to the CUSTOMER, but not title to the software.

B. Radio Shack also grants to CUSTOMER the right to reproduce the number of copies of the BISYNC-80/3780 Package required for CUSTOMER'S own personal, business, or other use. Each copy shall include Radio Shack's copyright notice.

C. Distribution of copies of the BISYNC-80/3780 Package other than as provided in this License, and resale of the BISYNC-80/3780 Package, are both strictly prohibited.

RADIO SHACK A DIVISION OF TANDY CORPORATION

U.S.A.: FORT WORTH, TEXAS 76102
CANADA: BARRIE, ONTARIO L4M 4W5

TANDY CORPORATION

AUSTRALIA
280-316 VICTORIA ROAD
RYDALMERE, N.S.W. 2116

BELGIUM
PARC INDUSTRIEL DE NANINNE
5140 NANINNE

U. K.
BILSTON ROAD WEDNESBURY
WEST MIDLANDS WS10 7JN

PRINTED IN U.S.A.