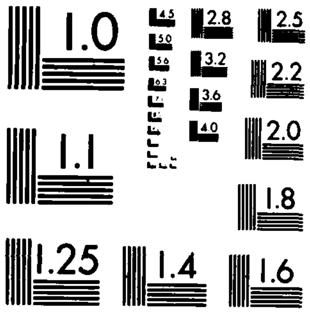
AD-A126 262 TEST MESSAGE GENERATOR AND CONTROLLER FOR AFSATCOM TESTING(U) MITRE CORP BEDFORD MA D O ALWINE MAR 83 MTR-8840 F19628-82-C-0001 1/2

UNCLASSIFIED F/G 17/2 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

(12)

ESD-TR-83-118 MTR-8840

TEST MESSAGE GENERATOR AND CONTROLLER FOR AFSATCOM TESTING

By
D. O. ALWINE

MARCH 1983

Prepared for
DEPUTY FOR STRATEGIC SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

Project No. 6340
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-82-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

[illegible]

JOSEPH E. MARDO, GS-13
Project Engineer

Max I. Miller Jr.

MAX I. MILLER, JR., Colonel, USAF
System Program Director
MILSTAR Program Office
Deputy for Strategic Systems

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-83-118	2. GOVT ACCESSION NO. AD-A126 262	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TEST MESSAGE GENERATOR AND CONTROLLER FOR AFSATCOM TESTING		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER MTR-8840
7. AUTHOR(s) D. O. ALWINE		8. CONTRACT OR GRANT NUMBER(s) F19628-82-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation Burlington Road Bedford, MA 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 6340
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Strategic Systems Electronic Systems Division, AFSC Hanscom AFB, MA 01731		12. REPORT DATE MARCH 1983
		13. NUMBER OF PAGES 128
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

AFSATCOM TESTING
AUTOMATED TESTING
MICROPROCESSOR-CONTROLLED
TEST MESSAGE GENERATOR

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The message controller in the D-91 Test Control Center (TCC) has been replaced with a new microprocessor-based unit in order to provide the increased flexibility needed for Air Force Satellite Communications (AFSATCOM) channel 1.5 testing.

This document describes the hardware and software used in the new test message generator and controller. Included are operating instructions, hardware description, software description, flow charts, assembly language listings, and a memory dump.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(*When Data Entered*)

UNCLASSIFIED

SECURITY CLASSIFICATION OF [illegible] PAGE(*When Data Entered*)

ACKNOWLEDGEMENT

This document has been prepared by the MITRE Corporation under Project 6340, Contract F19628-82-C-0001. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (concluded)

LIST OF TABLES

SECTION 1

INTRODUCTION AND OVERVIEW

Most Air Force Satellite Communications (AFSATCOM) system tests performed by the MITRE Corporation, including System Level Development Test and Evaluation (SLDT&E), used the previously developed version of a message controller available in the MITRE D-91 Test Control Center (TCC). This message controller connects to an AFSATCOM automatic send/receive (ASR) unit and transmits the message stored in the ASR buffer repeatedly. The operator selects the number of times the message is sent and the delay time between messages by means of front panel switches.

This technique was desirable during system level tests, as it permitted the message to originate in a standard AFSATCOM ASR and did not require the data stream to be stored or processed by any non-AFSATCOM equipment. System level testing has been completed for some time and the system was shown to perform satisfactorily.

However, since all test messages in a sequence are identical, this method of generating messages makes data reduction quite difficult. During an acquisition rate test, for instance, it would be advantageous to identify each message with a unique sequential number. Then, if a sequence of messages is sent and only a fraction of them received, it would be possible to determine whether the missed acquisitions are randomly distributed or grouped in bursts.

For the planned channel 1.5 testing, it was decided that a different approach should be taken to AFSATCOM test message generation. Plans were made to develop a new message controller based on a microprocessor to provide, as a minimum, the following general capabilities:

1. A moderate amount of user interactive capability via prompting.

2. The ability to insert a message number in the test message, if desired, to make each message unique.

3. The ability to insert a random data stream preceding the message to give a dual modem time to acquire when a test uses a regenerative channel in the laboratory without using a satellite.

4. The capability to simulate the signal from a satellite regenerative channel.

The Intel SBC-544 single board computer was chosen because (1) it has four serial input/output (I/O) ports, (2) the Tektronix 8002 development system owned by D-91 would support the 8085 chip, and (3) a spare Intel SBC 80/20 board which uses the same serial I/O chip as the SBC-544 was available for preliminary testing. This equipment made it possible to verify that the single board computer and the AFSATCOM modem (narrowband or wideband) can interface successfully.

To date, two of these message controllers have been constructed. Both were used extensively in the channel 1.5 testing and performed effectively.

Section 2 contains the operating instructions for the message controller. Sections 3 and 4 describe the hardware and software, respectively. A complete assembly language listing is provided in appendix A. Appendix B is a memory dump in Tektronix Microcomputer Development System (MDS) format.

SECTION 2

OPERATING INSTRUCTIONS

Operation of the message controller has been kept as simple as possible. The only front panel controls are an on/off switch and a reset button. The only connections to the unit are AC power, and cables to the AFSATCOM modem I/O connector and a terminal device. A Texas Instruments (TI) model 765 ASR serves as the terminal device. However, almost any terminal device could be used if it has an Electronic Industries Association (EIA) RS-232 interface and is set for the following:

1. An 8-bit word (7 bits plus parity)

2. Odd parity

3. 300 baud

4. 1 start bit and 1 stop bit

5. Full-duplex operation

The primary function of the message controller is to transmit a message many times in repetitive message testing. The unit also has an ASR emulation mode in which it can partially emulate the operation of an AFSATCOM ASR. Also provided is a special test mode which causes a jump to location 0800 in program memory. This location is the address of an empty programmable read-only-memory (PROM) socket, and permits the easy addition of some other user-defined test function at a later date.

2.1 EXTERNAL CONNECTIONS

Direct connection to an AFSATCOM modem can be made with a cable wired as shown in table 2-1. Connection to a modem via the TCC patch panel requires a cable wired according to table 2-2.

The cable from the data terminal is plugged into port Ø on the message controller. This is an RS-232 I/O port. The pin connections to port Ø are listed in table 2-3. Figure 2-1 is a diagram of the external connections to the message controller.

Table 2-1

Connections to Modem

Modem Connector MS27484T16F35S		RS-232 Female Conn. (Rear Pannel J2)
25	Receive (RX) Data	3
37	Ground	1
26	RX Clock	17
8	Transmit (TX) Data	2
20	Ground	7
9	TX Clock	15
21	TX Enable	4

Table 2-2

Connections to TCC Patch Panel

Patch Panel Cinch 57-40240		RS-232 Female Conn. (Rear Panel J2)
1	RX Data	3
2	Ground	1
3	RX Clock	17
13	TX Data	2
14	Ground	7
15	TX Clock	15
17	TX Enable	4

Table 2-3

Pin Assignments in RS-232C Interface
to Data Terminal Device

Pin #	Assignment
1	Chassis Ground
2	TX Data
3	RX Data
4	Request to Send (RTS)
5	Clear to Send (CTS)
6	Data Set Ready (DSR)
7	Signal Ground
13	Data Carrier Detect (DCD)
20	Data Terminal Ready (DTR)

2.2 INITIATING A REPETITIVE MESSAGE TEST

The flowchart of figure 2-2 shows the step-by-step procedure for initiating a repetitive message test. Refer to this flowchart throughout section 2.2.

After making the connections to the AFSATCOM modem and the data terminal, turn on the data terminal. The TI-765 will respond by printing:

```
Ready *B
PROM 7
```

The above statement applies only to TI-765 operation, since most terminals do not print anything when turned on.

After turning on the data terminal, turn on the message controller, which then prints an operator prompt message:

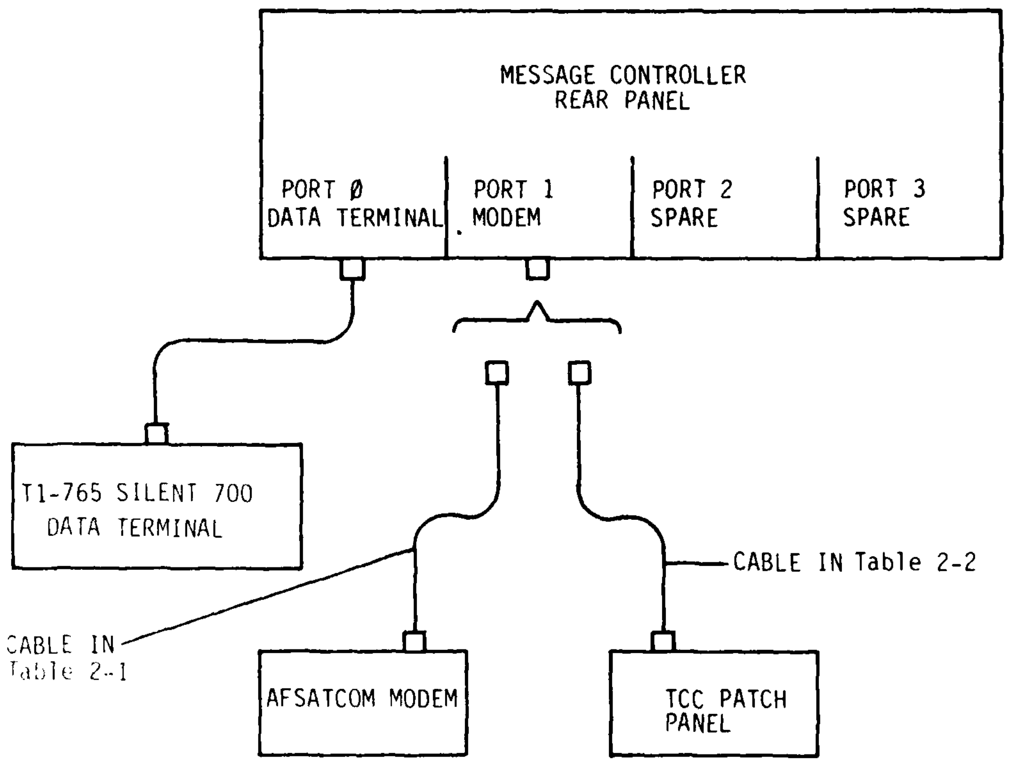


Figure 2-1. External Connections to Message Controller

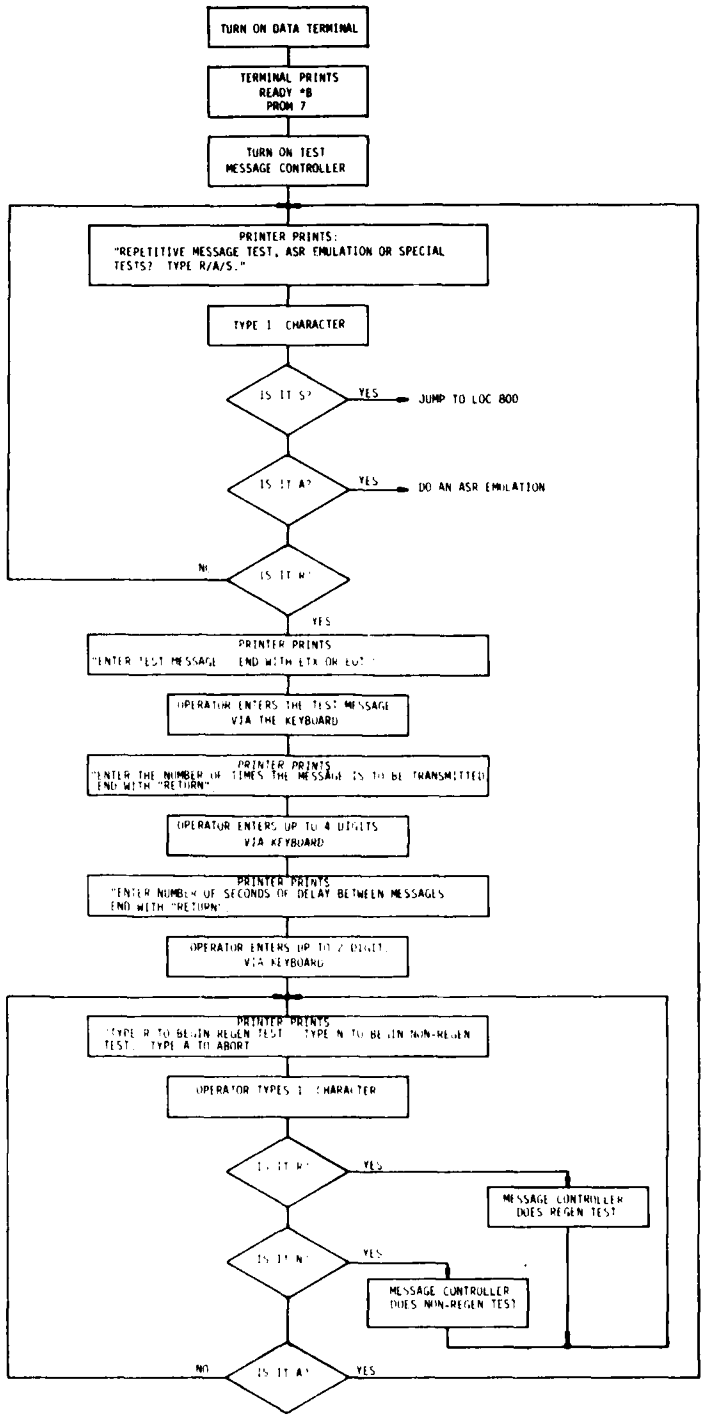

TURN ON DATA TERMINAL
TERMINAL PRINTS READY *B PROM 7
TURN ON TEST MESSAGE CONTROLLER
PRINTER PRINTS: "REPETITIVE MESSAGE TEST, ASR EMULATION OR SPECIAL TESTS? TYPE R/A/S."
TYPE 1 CHARACTER
IS IT S?
YES
JUMP TO LOC 800
IS IT A?
YES
DO AN ASR EMULATION
NO
IS IT R?
YES
PRINTER PRINTS "ENTER TEST MESSAGE END WITH ETX OR EOT"
OPERATOR ENTERS THE TEST MESSAGE VIA THE KEYBOARD
PRINTER PRINTS "ENTER THE NUMBER OF TIMES THE MESSAGE IS TO BE TRANSMITTED END WITH "RETURN"
OPERATOR ENTERS UP TO 4 DIGITS VIA KEYBOARD
PRINTER PRINTS "ENTER NUMBER OF SECONDS OF DELAY BETWEEN MESSAGES END WITH "RETURN"
OPERATOR ENTERS UP TO 2 DIGITS VIA KEYBOARD
PRINTER PRINTS "TYPE R TO BEGIN REGEN TEST TYPE N TO BEGIN NON-REGEN TEST. TYPE A TO ABORT
OPERATOR TYPES 1 CHARACTER
IS IT R?
YES
MESSAGE CONTROLLER DOES REGEN TEST
IS IT N?
YES
MESSAGE CONTROLLER DOES NON-REGEN TEST
NO
IS IT A?
YES

Figure 2-2. Procedure for Initiating a Repetitive Test

```
REPETITIVE MESSAGE TEST, ASR EMULATION, OR SPECIAL TEST?
TYPE R/A/S
```

To initiate a repetitive message test, the operator types an upper case "R". Typing any character other than upper case "R", "A", or "S" will cause the prompt to be repeated. The "A" and "S" are discussed in section 4. Typing "R" causes the controller to enter the repetitive message test mode.

2.2.1 Entering a Test Message

After the operator types the upper case "R", the computer will respond with the prompt:

ENTER TEST MESSAGE. END WITH <u>ETX</u> OR <u>EOT</u>

At this point, the operator enters the message to be used in the test. Enough memory is available for over 16,000 characters, which is more than enough for any practical test. The test message must be terminated with an <u>ETX</u> (control C) or an <u>EOT</u> (control D). The difference between <u>ETX</u> and <u>EOT</u> is discussed in sections 2.3.4 and 2.3.5.

The operator can correct a typing error by backspacing the appropriate number of characters and retyping. On most data terminals, control H will generate an American Standard Code for Information Interchange (ASCII) backspace character. Some terminals also have a backspace key. However, many backspace keys only back up the printhead (or cursor on a cathode ray tube (CRT) terminal) and do not transmit the <u>BS</u> character. This is the case with the TI-765. To backspace both the memory and the printer when using the TI-765, use control H, do not use the backspace key.

A second way to correct an error is to type the <u>RS</u> character (Control ".") This causes the prompt (ENTER TEST MESSAGE....) to be reprinted and allows the operator to re-enter the entire test message. The operator always must end the message with an <u>ETX</u> or an <u>EOT</u>.

2.2.2 Entering the Number of Test Message Transmissions

As soon as the operator terminates entry of the test message (by typing <u>ETX</u> or <u>EOT</u>), the computer will respond with another prompt:

```
ENTER THE NUMBER OF TIMES THE MESSAGE IS TO BE TRANSMITTED.
END WITH "RETURN."
```

When this prompt has been printed, the operator can enter any number up to 9999 via the keyboard. Actually, any number of digits may be typed, but the computer only "looks" at the last four typed prior to the carriage return. This enables an operator to correct a mistake by merely typing several zeros, followed by the correct number and a carriage return. If a character other than a digit from 0 to 9 is typed, the computer will prompt the operator to begin entering the parameter again.

2.2.3 Entering the Delay Time Between Messages

When the number of test messages has been entered, the computer will prompt the operator by printing:

```
ENTER THE NUMBER OF SECONDS OF DELAY BETWEEN MESSAGES.
END WITH "RETURN."
```

The operator can now enter any whole number of seconds up to 99 via the keyboard. Again, any number of digits may be typed, but the processor uses only the last two. If a character other than a digit from 0 to 9 is entered, the computer will prompt the operator to begin the process of inputting the delay time again.

2.2.4 Selecting Regenerative or Non-Regenerative Testing

As soon as the delay time has been entered, the computer will print:

```
TYPE R TO BEGIN REGEN TEST
TYPE N TO BEGIN NON-REGEN TEST
TYPE A TO ABORT
```

Typing an "A" will cause the processor to return to the very beginning of the program. Typing "R" or "N" will initiate the transmission of test messages, as discussed in the next two paragraphs.

2.2.4.1 Non-Regenerative Testing

When the operatoı initiates the test message transmission by typing "N", the result is like using the old message controller (with the addition of features that are discussed in section 2.3). That is, the I/O transmit (TX) enable line to the modem is set equal to a logic "1", the modem clock shifts the message out of the computer one bit at a time, and then the I/O TX enable is set equal to a logic "0" when the message transmission is completed. This cycle repeats until the message has been sent the specified number of times.

2.2.4.2 Regenerative Testing

Initiating the test by typing "R" simulates the output of a satellite regenerative channel. This mode would be used only in a laboratory set-up where the transmitting modem is sending directly to the receiving modem without a satellite and must simulate the output of a satellite regenerative channel. This is done by keeping the transmitter keyed on for the duration of the test, with preambles and postambles inserted by the computer. The delay time between messages is filled with random data.

As soon as the operator types "R", the following sequence occurs:

- The transmitting modem is keyed up. (I/O TX enable is set equal to a logic "1.")
- A 10-character random data table is transmitted N times. N is equal to the number of seconds of delay between messages, which the operator entered previously.
- A WU SYN SYN preamble is transmitted.
- The test message is transmitted.
- An ETX ETX ETX ETX even parity postamble is transmitted.
- The 10-character set of random characters is transmitted another N times.
- The sequence of preamble, message, postamble, and random data is repeated until the message has been sent the required number of times.
- The transmitter is turned off by setting the I/O TX enable line to equal logic "0".
- The prompt "TYPE R TO BEGIN REGEN TEST, TYPE N TO BEGIN NON-REGEN TEST, TYPE A TO ABORT" is printed again.

At this point, typing "R" would repeat the entire test again, typing "N" would repeat the test but it would be non-regenerative (as described in paragraph 2.2.4.1), and typing "A" would abort the test and ask the operator to select a repetitive message test, an ASR emulation, or a special test by typing "R", "A", or "S".

2.2.5 Halting a Test in Progress

When a test is being conducted, it is sometimes desirable to halt the test before the specified number of messages has been transmitted. Three ways in which a test in progress may be halted are discussed in the following paragraphs.

2.2.5.1 Temporary Suspension of a Test in Progress

The operator can temporarily suspend a test by depressing the space bar. The message being transmitted will continue until the entire message has been sent. However, after the pause normally inserted between messages, the transmission will be suspended until the space bar is depressed again. While the test is suspended, the transmitter will remain off if the test is non-regenerative. If the test is a regenerative channel simulation, synchronous idle characters will be sent while the test is suspended. When testing resumes, message transmission will start exactly where it left off.

This feature is useful if, for instance, the operator discovers during a test that the receiving printer is out of paper. The test could be suspended while the paper roll is being changed and then resumed at exactly the point where it left off. The number of messages sent before the suspension plus those sent afterwards will equal the desired number of messages in the test sequence.

If the operator types "R" or an "A" while testing is suspended, the test will be restarted or aborted.

2.2.5.2 Restarting a Test in Progress

If the operator types an upper case "R" (restart) while the test is in progress, the program returns to the prompt:

```
TYPE R TO BEGIN REGEN TEST
TYPE N TO BEGIN NON-REGEN TEST
TYPE A TO ABORT
```

This permits a test to be terminated prematurely and restarted from the beginning. This feature would be useful if an operator inadvertently initiated a regenerative test instead of a non-regenerative, or if the equipment was misadjusted, making it desirable to begin the test again.

2.2.5.3 Aborting a Test in Progress

If the operator types an upper case "A" while the test is in progress, the message being transmitted will continue until it has been completely transmitted. After the pause interval between messages has been completed, the program will begin execution at the prompt:

```
REPETITIVE MESSAGE TEST, ASR EMULATION, OR SPECIAL TEST?
TYPE R/A/S
```

At this point, the entire test has been aborted and the test message and all parameters must be reentered.

2.3 SPECIAL FUNCTIONS

A test message may contain any valid ASCII characters. With five exceptions, all characters entered into the test message by the operator are transmitted without change to the modem. The five exceptions are given in table 2-4. ETX is used to terminate a message, exactly as it is normally used in the AFSATCOM system. The other four characters were deliberately chosen because they are not used anywhere in the AFSATCOM system. The uses of these special characters are described in the subsequent paragraphs.

Table 2-4

Special Characters

ASCII Character	Key
SUB	CTRL Z - inserts message number
US	CTRL / - outputs next character with even parity
FS	CTRL , - inserts random data in message
ETX	CTRL C - ends message with ETX sent twice
EOT	CTRL D - ends message with no ETX

2.3.1 Inserting a Message Number in the Message

The number of the message can be transmitted as part of the test message by using the SUB (control Z) character in the test message. SUB will not be transmitted as part of the message. Instead, the computer will transmit a three-digit number. This number would be ØØØ in the first message, 001 in the second message, 099 in the 100th message, etc. The message number can be at the beginning of the message, at the end, or anywhere in the middle. This number can also be used more than once in the same message.

Note that the message number which is printed consists of the three least significant digits of a four-digit count. In the unlikely event that the number of times the message is to be transmitted exceeds 1000, the printed count would recycle back to ØØØ after 999. However, the message count internal to the computer will keep accurate count to 9999.

2.3.2 Inserting an Even Parity Character in the Test Message

The operator can insert an even parity character by typing the US (control /) character before the desired even parity character. US is not transmitted; this character affects only the parity bit of the next character of the message. For instance, an even parity character could be used to simulate exactly a time division multiplex (TDM) frame synch message. All frame synch messages begin with an even parity lower case i. The operator would enter this as USi followed by the rest of the sync message. No space character is permitted between the US and the i. If a space character were inserted, the space, not the i, would be sent with even parity.

2.3.3 Inserting Random Toggling in a Test Message

The operator can insert N seconds of random data in a test message by typing FS (control ","), followed by two digits. Neither the FS nor the two digits are transmitted. Instead, the two digits are read and set equal to N. The 10 character random table is then sent N times.

This feature is used primarily when the operator is transmitting data to a dual modem on a regenerative channel in the laboratory without a satellite. Several seconds of random data inserted at the beginning of the message give the dual modem a chance to acquire the signal. The operator must type the preamble into the test message following the toggling.

For example, to see if a dual modem will acquire in 50 characters (400 bits), the test message would be as follows:

FS05 WU SYN SYN - - - - TEST MESSAGE - - - -ETX.

The transmitted message would consist of the random table sent five times (50 characters total), followed by the WU SYN SYN and the rest of the test message.

Note that the two digits must immediately follow the FS character. No spaces are allowed between the FS and the number.

2.3.4 Terminating Messages Normally

The operator normally terminates message entry by typing ETX. Transmission of the test message terminates at the ETX. The ETX is sent twice to duplicate the action of the AFSATCOM ASR.

2.3.5 Terminating a Message Without an ETX

Message transmission can be terminated without the transmission of an ETX if the operator types the EOT (control D) character. EOT is not transmitted. Transmission terminates when the last bit of the character preceding the EOT has been sent.

2.4 AFSATCOM ASR EMULATION MODE

Typing "A" in response to the prompt "REPETITIVE MESSAGE TEST, ASR EMULATION, OR SPECIAL TEST? TYPE R/A/S" will cause the processor to enter the ASR emulation mode.

This prompt is printed immediacely upon power up, each time a test is aborted, and whenever the front panel "reset" button is depressed.

The ASR emulation mode was included primarily as a means of testing the receive portion of the hardware interface and verifying that the RCVMSG subroutine will run. This mode is only a partial emulation of the AFSATCOM ASR, and should be refined if a need for such operation is defined. Enhancement would be fairly simple and straightforward.

Once the ASR emulation mode has been entered, the only response will be a carriage return/line feed immediately after the "A" is typed. The processor then enters a "wait" state, waiting either for a message to be entered via the keyboard, or for a message to be received via the AFSATCOM modem.

While in the "wait" state, if the operator types any character except escape ESC, the processor enters the compose/edit mode. In this mode, all typed characters except BS (control H) are stored in a buffer. BS causes a backspace so that the operator can type over mistakes. When an ETX is typed, the processor returns to the "wait" state. While the processor is in the compose/edit mode, any message received from the modem will be ignored.

While the processor is in the "wait" state, typing ESC will cause the message in the buffer to be transmitted. This can be done repeatedly. Thus, the escape key acts in a manner similar to the "AUTO XMT" key on the AFSATCOM ASR. While a message is being transmitted, any received message will be ignored.

During the "wait" state, any message received will be printed, character by character, as it is received. Receiving a message does not disturb the transmit buffer. After a message is received, the processor returns to the "wait" state, ready to receive another message, to compose another message, or to retransmit a previously composed message.

To exit from the ASR emulation mode, the operator must push the front panel "reset" switch.

2.5 SPECIAL TEST MODE

Typing "S" in response to the prompt "REPETITIVE MESSAGE TEST, ASR EMULATION OR SPECIAL TEST? TYPE R/A/S" will cause the program execution to jump to location 0800. This is the address of an empty socket available for a 2716 PROM chip programmed with a user-defined test sequence.

If a need is defined for a test mode different from what is currently provided, the necessary program can be written and programmed into a 2716 PROM chip. Any new program installed in this manner can make full use of existing subroutines. One example of a supplemental program at location 0800 would be one to process received messages.

SECTION 3

HARDWARE DESCRIPTION

The message controller is built around an Intel SBC-544 single board computer. This computer has four serial I/O ports and three parallel I/O ports. The serial I/O ports are RS-232 compatible and each is available on a separate card edge connector.

A block diagram of the message controller is shown in figure 3-1. The data terminal (TI-765) connects directly to serial port 0. Serial port 1 connects through an interface board to the AFSATCOM modem. Serial ports 2 and 3 are not currently used.

The front panel contains a reset button which forces a power-up reset condition, an on/off switch, a power on indicator, and six indicator lights with the following functions:

1. TX enable - indicates that the I/O TX enable line is equal to a logic "1". (Transmitter is keyed on.)

2. TX bit clock - indicates that the modem bit clock is running.

3. TX data - indicates that transmit data is flowing from the computer to the modem.

4. RX clock - indicates that the modem is receiving a message.

5. RX data - indicates the flow of received data from the modem to the comptuer.

6. REGEN - indicates that the processor is simulating the output of a regenerative satellite channel.

The single board computer plugs into a card cage, which is bolted to a rack-mountable tray along with a power supply, two cooling fans, the front panel, and the interface circuit board.

The only external connections needed are AC power, a connection to a data terminal, and a connection to an AFSATCOM modem. Refer to section 2.1 for a detailed description of these connections.

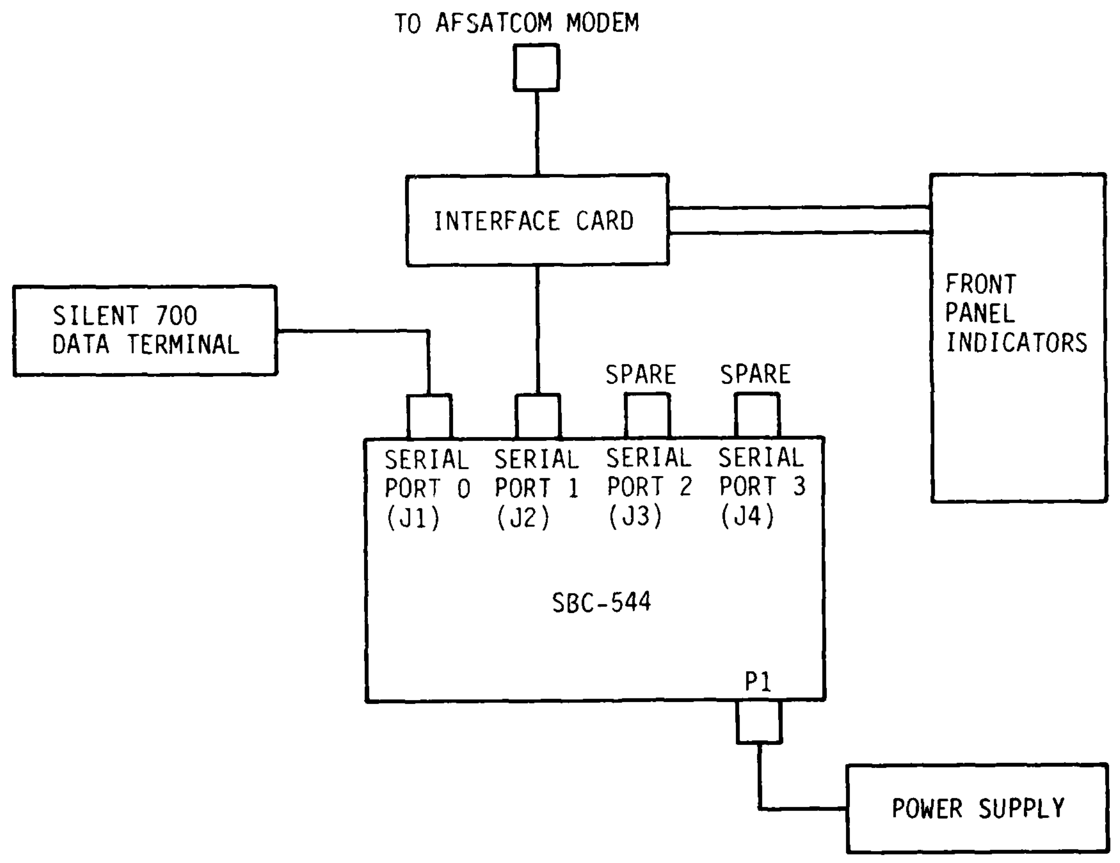


Figure 3-1. Block Diagram of Message Controller

3.1 INTEL SINGLE BOARD COMPUTER

The Intel SBC-544 single board computer uses an 8085 microcomputer chip, four 8251A universal synchronous-asynchronous receiver-transmitter (USART) chips, an 8155 parallel I/O chip, two 8253 timer chips, an 8259 programmable interrupt controller, 16 kilobytes of dynamic random access memory (RAM), and up to 8 kilobytes of read-only memory (ROM).

The single board computer has extensive interrupt capability. The present system does not use interrupts. However, it would be worthwhile to modify the software to utilize interrupts in case the system needs to handle several I/O functions simultaneously. Examples of simultaneous I/O are full-duplex operation and handling messages to more than one modem.

For details about addressing I/O ports, programming the baud rate, computer instruction set, etc., refer to the SBC-544 Intelligent Communications Controller Board Hardware Reference Manual, Manual Order Number 9500616B, Intel Corporation, 1978.

3.1.1 Jumper Options

The Intel single board computer has a number of options that the user can select by means of jumper wires attached to wire-wrap terminals. Most of the connections were left in the default configuration as supplied by the factory. The changes to the factory supplied jumpers are shown in table 3-1.

Table 3-1

Changes to Factory Supplied Jumper Connections

Purpose:	Change Needed:
Select 2716 ROM	Connect 38-39, 40-41
Select external clocks for USART 1 (For AFSATCOM modem)	Remove 9-11, 13-14 Connect 9-10, 12-13

3.1.2 Modifications to the Single Board Computer

Two modifications were made to the single board computer. The first, described in paragraph 3.1.2.1, was necessary to provide a data carrier detect (DCD) signal to the TI-765 data terminal. The second modification, described in paragraph 3.1.2.2, was needed to provide a synch detect signal to the USART chip to establish receive character synch.

3.1.2.1 Modification to Provide DCD to the Data Terminal

The four serial I/O ports on the SBC-544 are each brought out to a separate card edge connector. The pinouts are such that a connection to a standard Bell modem can be made using a 25 conductor ribbon cable with a 26 pin card edge connector on one end and an EIA standard 25 pin male "D" type connector on the other. A drawing of the cable is shown in figure 3-2.

When a data terminal (ASR) is connected to a data set (Bell modem), the transmit data flows out of the data terminal on pin 2 of the interfacing connector. This same data flows into the modem on pin 2. Likewise, receive data flows out of the modem and into the ASR via pin 3 of the mated connector pair.

As discussed earlier, the computer board shipped from the factory is configured for use as a data terminal. That is, data from the computer flows out of pin 2 of the RS-232 connector in figure 3-2. To connect a terminal device such as the TI-765 to the computer, the computer I/O port must be reconfigured to make the computer look like a data set (Bell modem), i.e., all outputs must become inputs and all inputs must become outputs.

Intel designed this board so that making such a change is almost convenient. Refer to pages 5-21/5-22 of the SBC-544 Instruction Manual. Each serial I/O port has an associated dual in-line package (DIP) connector with jumper plug installed. By removing this plug and installing another with different jumpers, it is possible to transpose some of the lines. However, some of the I/O "handshake" lines do not pass through the jumper plug. One of these is the DCD line. DCD is an input to the SBC-544, although its use is optional. No provision was made to make DCD an output. Use of DCD is not optional to the TI-765 nor to many other terminals; DCD must be present or the terminal will not operate.

To keep the computer compatible with any terminal device, we modified the computer board rather than the wiring in the terminal I/O cable. Refer to the SBC-544 Instruction Manual, page 5-17; the DCD signal for port 0 arrives in pin 16 of J1. The signal goes

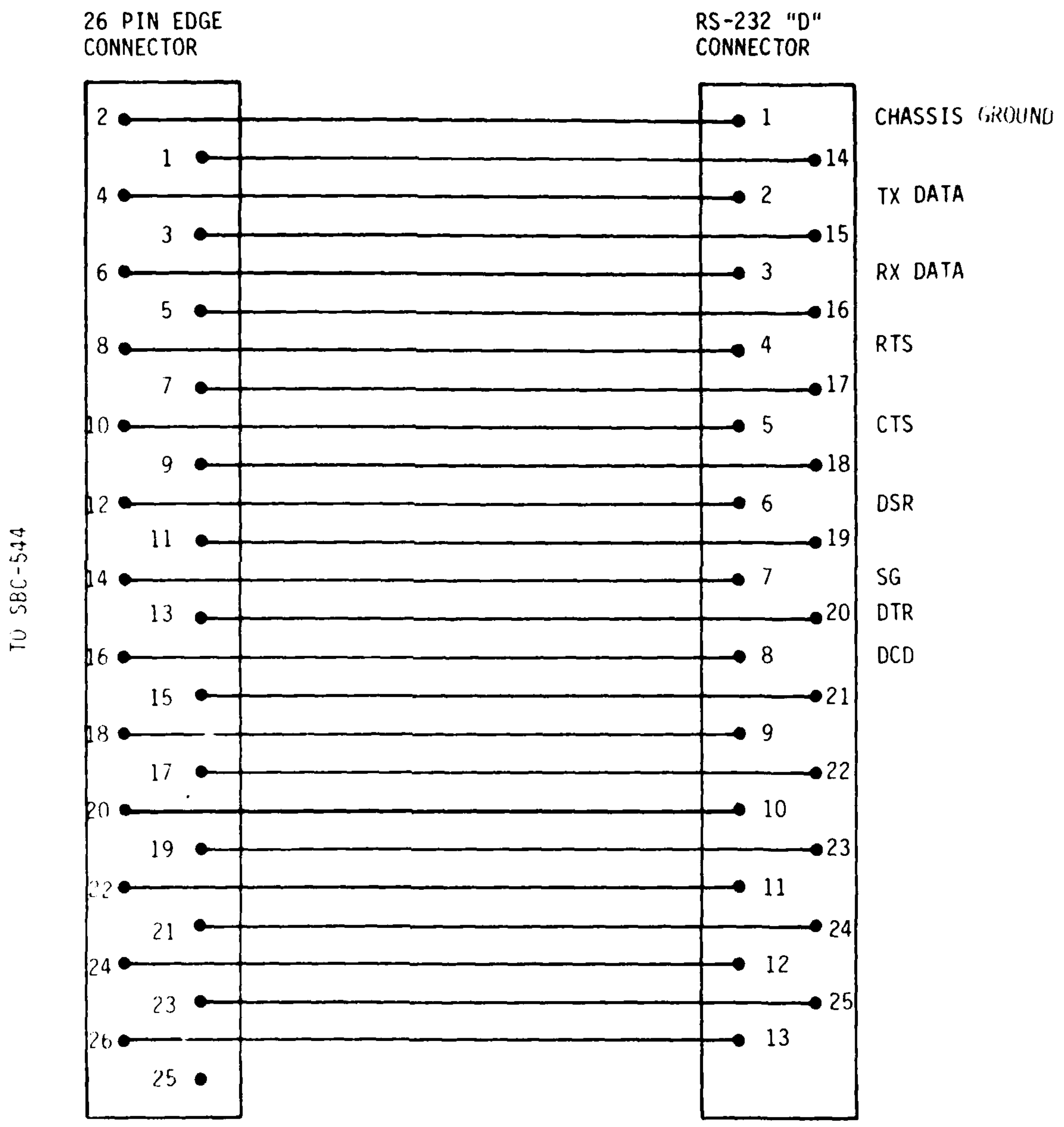


Figure 3-2. Serial I/O Connections to the Single Board Computer with Factory Supplied Jumpers

directly to pin 10 of A11, a 1489 line receiver input. By soldering a jumper between pins 10 and 14 (the +5V V_{CC} pin) of A11, +5 V can be made to appear continuously on pin 16 of card edge connector J1 whenever the power to the single board computer is on. This voltage level is sufficient to satisfy the requirements of the TI-765, or any other terminal device.

3.1.2.2 Modification to Establish Receive Character Synch

When the serial I/O chips are used in the asynchronous mode, the first bit is always a start bit. Thus, character synch is never a problem. In the synchronous mode, there are no start bits, no stop bits, and no pauses between characters. Therefore, some means must be provided to establish character synchronism. Intel designed the serial I/O chip with provision for establishing synch in one of two ways.

The first technique for establishing character synch utilizes synch words in the data stream. The I/O chip continually monitors the incoming serial data, and begins assembling the incoming bits into words when a synch word is detected. Since the AFSATCOM modem strips off the WU SYN SYN preamble, and since no other unique sequence is sent at the beginning of an AFSATCOM message, this synch technique is not usable with AFSATCOM and is not discussed further.

The second means of establishing character synch on incoming data is to apply a logic level to pin 16 of the serial I/O (USART) chips. A transition from a logic "1" to a logic "0" will cause the chip to begin assembling the bits into words, beginning with the next transition of the bit clock to the active state. This signal can be derived easily from the bit clock (see section 3.2). However, while the serial I/O chip provides for external synch, the single board computer does not utilize this provision. No connection was made to pin 16 for any of the USART chips. Therefore, a modification had to be made to the board to bring the synch signal to pin 16 of the port 1 USART chip.

Refer to figure 3-3a (reproduced from the SBC-544 Instruction Manual). Pin 5 of J2, labeled SRXD, is connected to jumper plug W2 pin 11, through the jumper plug to pin 8, and from there, off the edge of the paper to sheet 6 (figure 3-3b). SRXD1 appears at pin 13 of A5, a 1489 line receiver chip in figure 3-3b. The corresponding output is A5 pin 11. Soldering a wire from A5 pin 11 to A19 pin 16 (see figure 3-3a again) provides a means of bringing the SYNDET signal from J2 pin 5 to the USART chip (A19) pin 16, as long as the plug in W2 has a jumper between pins 8 and 11. The SYNDET signal originates in the computer to modem interface.

3.2 COMPUTER-TO-MODEM INTERFACE

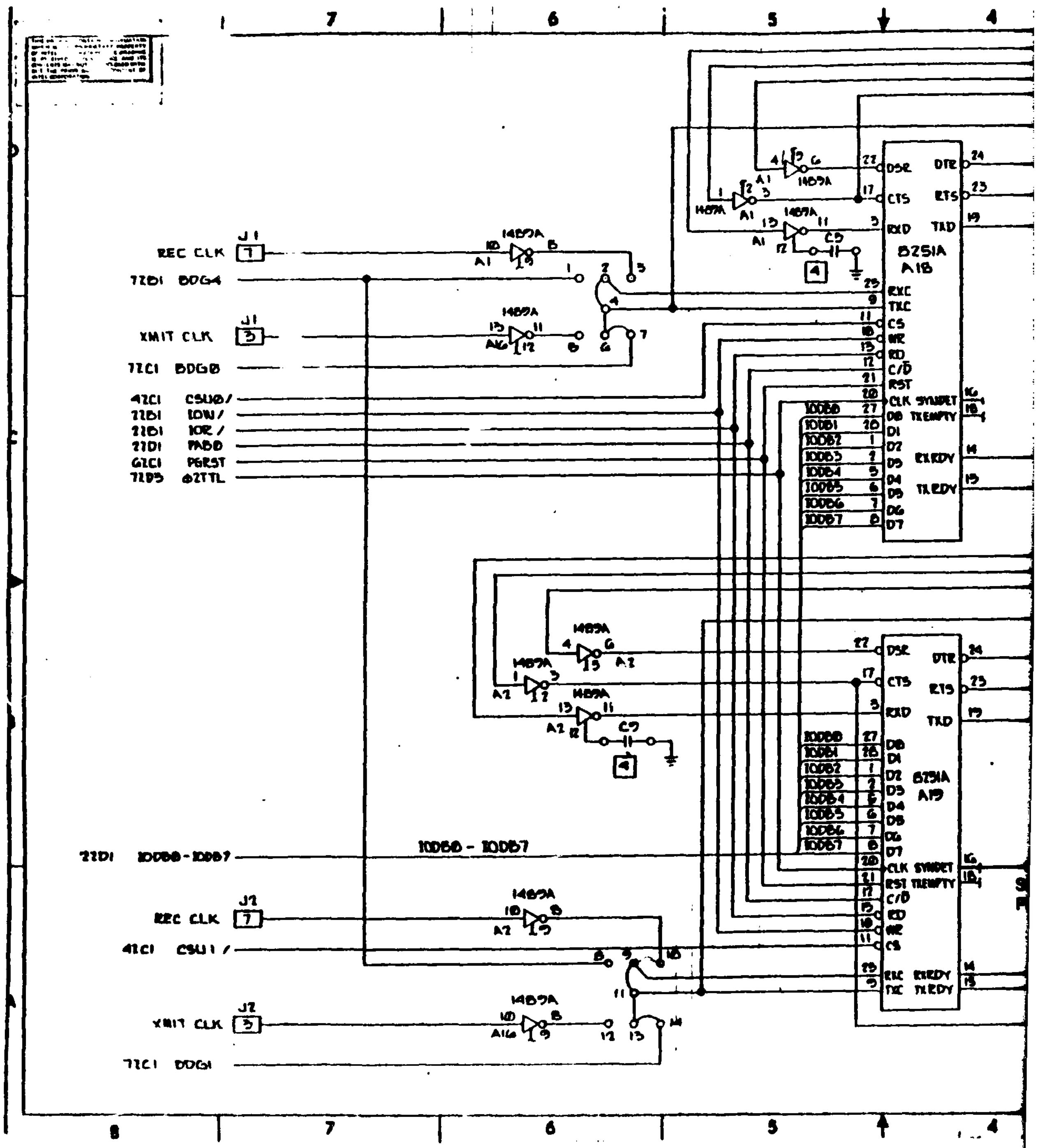
The computer-to-modem interface was constructed at MITRE and serves the following functions:

- Inserts an extra transmit clock pulse at the beginning of each transmitted message.
- Generates a SYNDET pulse. SYNDET establishes bit synch in the Intel serial I/O chip. SYNDET also is used to indicate to the computer whether or not the bit clock is running, and thus, whether or not a message is being received.
- Delays the rising edge of the first receive clock pulse slightly so that the SYNDET pulse can stabilize before the first clock transition occurs.
- Limits the $\pm$12 V RS-232 compatible signal levels from the computer to $\pm$6 $\pm$1 V for the AFSATCOM modem, as required by MIL-D-188C.
- Debounces the front panel "reset" switch, to provide a clean reset pulse to the computer.
- Provides drivers for the front panel indicator lamps.

Each of these functions is described in the following paragraphs. The schematic of the interface is shown in figure 3-4. The cabling harness connecting the single board computer, interface board, modem connector, and front panel is shown in figure 3-5.

3.2.1 Transmit Data Timing

Transmission of a message via AFSATCOM begins with changing the level on the I/O TX enable line from -6 V to +6 V. The modem then turns on the transmitter and sends a WU $\underline{\text{SYN SYN}}$ preamble. After the preamble has been sent, the modem turns on the external TX clock. The device providing the TX data must provide a new data bit each time the modem clock rises from -6 V to +6 V. The modem samples the data bit on the falling edge of the clock pulse. When all of the data bits comprising a message have been transferred into the modem, the I/O TX enable line must be made to transition from +6 V to -6 V. This causes the modem to turn off the transmitter after sending a postamble consisting of four even-parity $\underline{\text{ETX}}$ characters.



REC CLK
J1
7
72B1 BDG4
XMIT CLK
J1
3
72C1 BDG0
42C1 CSU0/
72B1 IOW/
72B1 IOR/
72D1 PAB0
62C1 PGRST
72B5 Ø2TTL
1489A
8251A
A18
8251A
A19
72D1 IODB0-IODB7
IODB0 - IODB7
REC CLK
J2
7
42C1 CSU1/
XMIT CLK
J2
3
72C1 BDG1
C5
C7
4
DSR
CTS
RXD
RXC
TXC
CS
WR
RD
C/D
RST
CLK
DTR
RTS
TXD
SYNDET
TXEMPTY
RXRDY
TXRDY

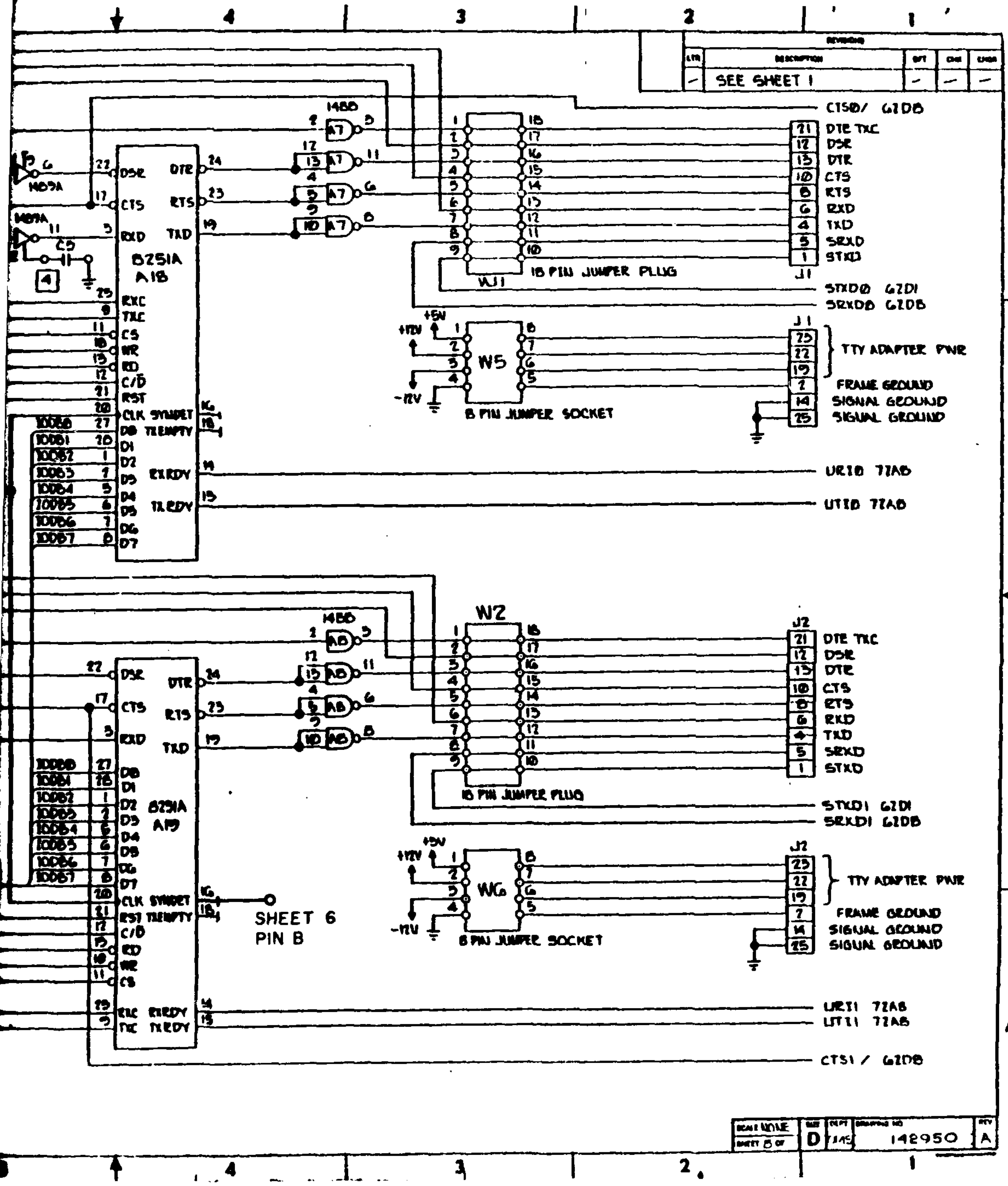
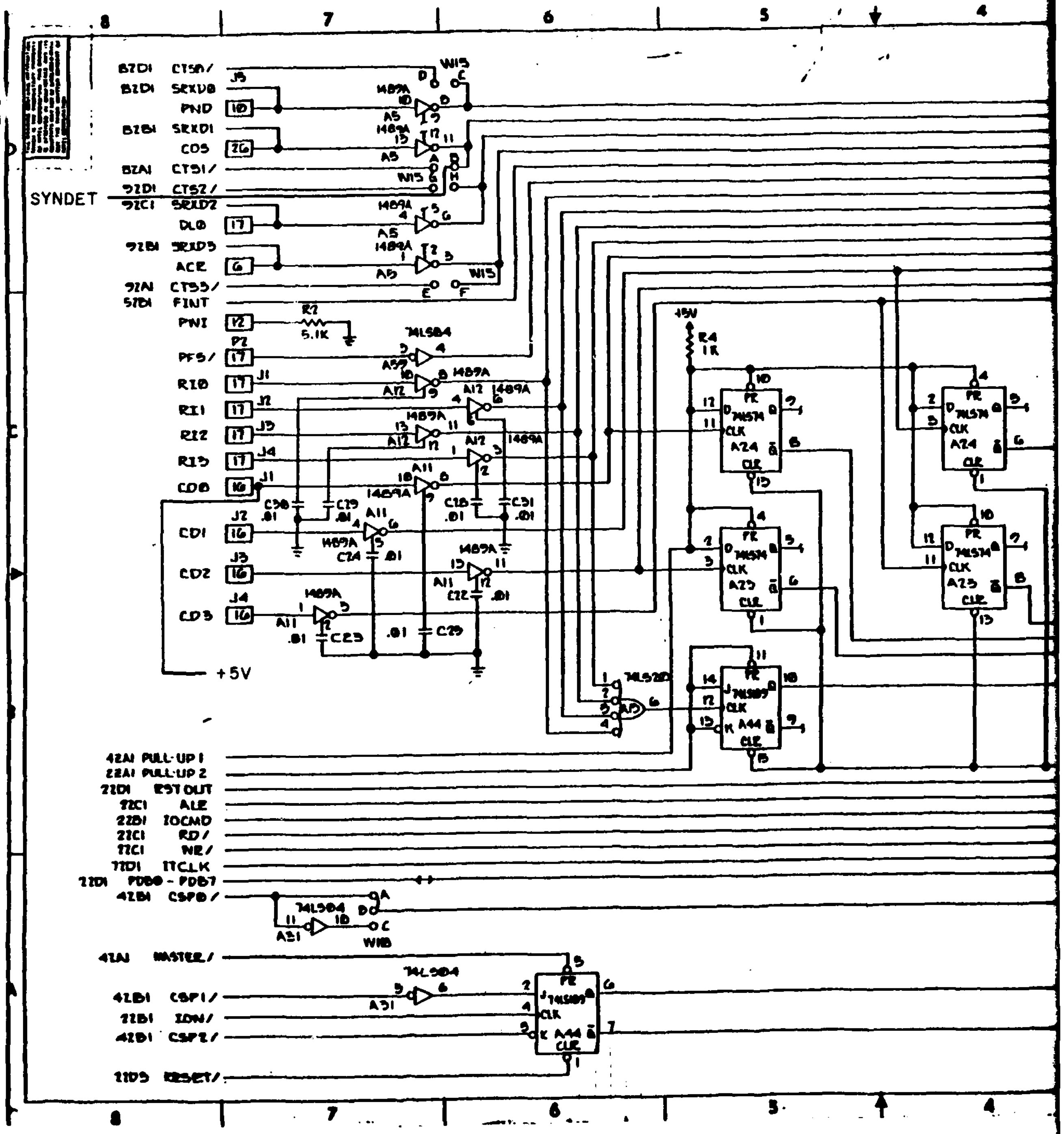


Figure 3-3a. Interface Schematic, Sheet 8

2



8
7
6
5
4
SYNDET
B2D1 CTS0/
B2D1 SRXD0
PND
B2B1 SRXD1
CDS
B2A1 CTS1/
92D1 CTS2/
92C1 SRXD2
DL0
92B1 SRXD3
ACE
92A1 CTS3/
57D1 FINT
PWI
R2 5.1K
PFS/
RI0
RI1
RI2
RI3
CD0
CD1
CD2
CD3
+5V
W15
1489A
A5
A11
A12
74LS04
R4 1K
74LS74
A24
A23
74LS109
A44
74LS20
A13
C30 .01
C29 .01
C18 .01
C31 .01
C24 .01
C22 .01
C23
C25
42A1 PULL-UP 1
22A1 PULL-UP 2
22D1 RST OUT
22C1 ALE
22B1 IOCMD
22C1 RD/
22C1 WR/
22D1 TTCLK
22D1 PDB0 - PDB7
42B1 CSP0/
A31
W18
42A1 MASTER/
42B1 CSP1/
22B1 IOW/
42B1 CSP2/
22D3 RESET/

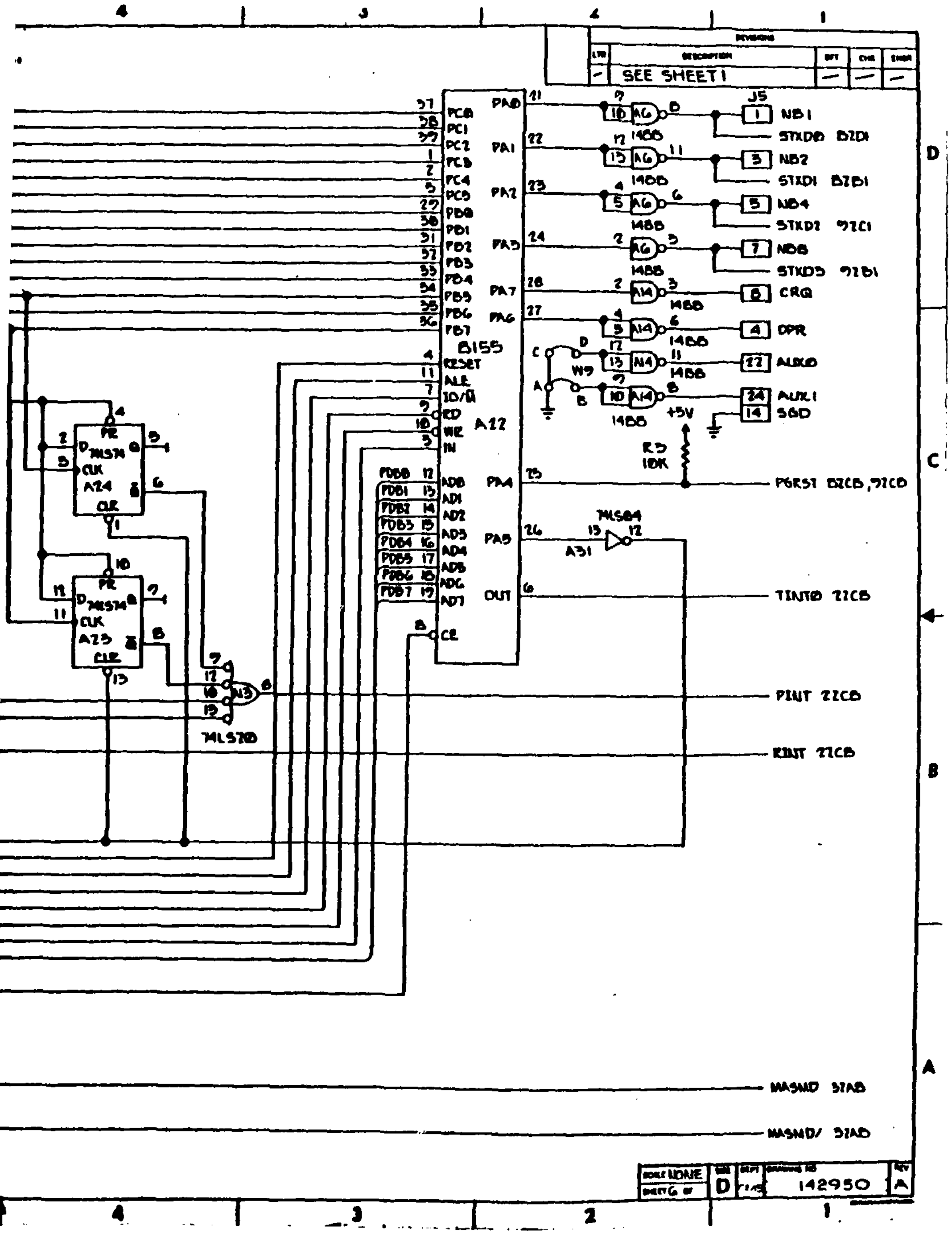
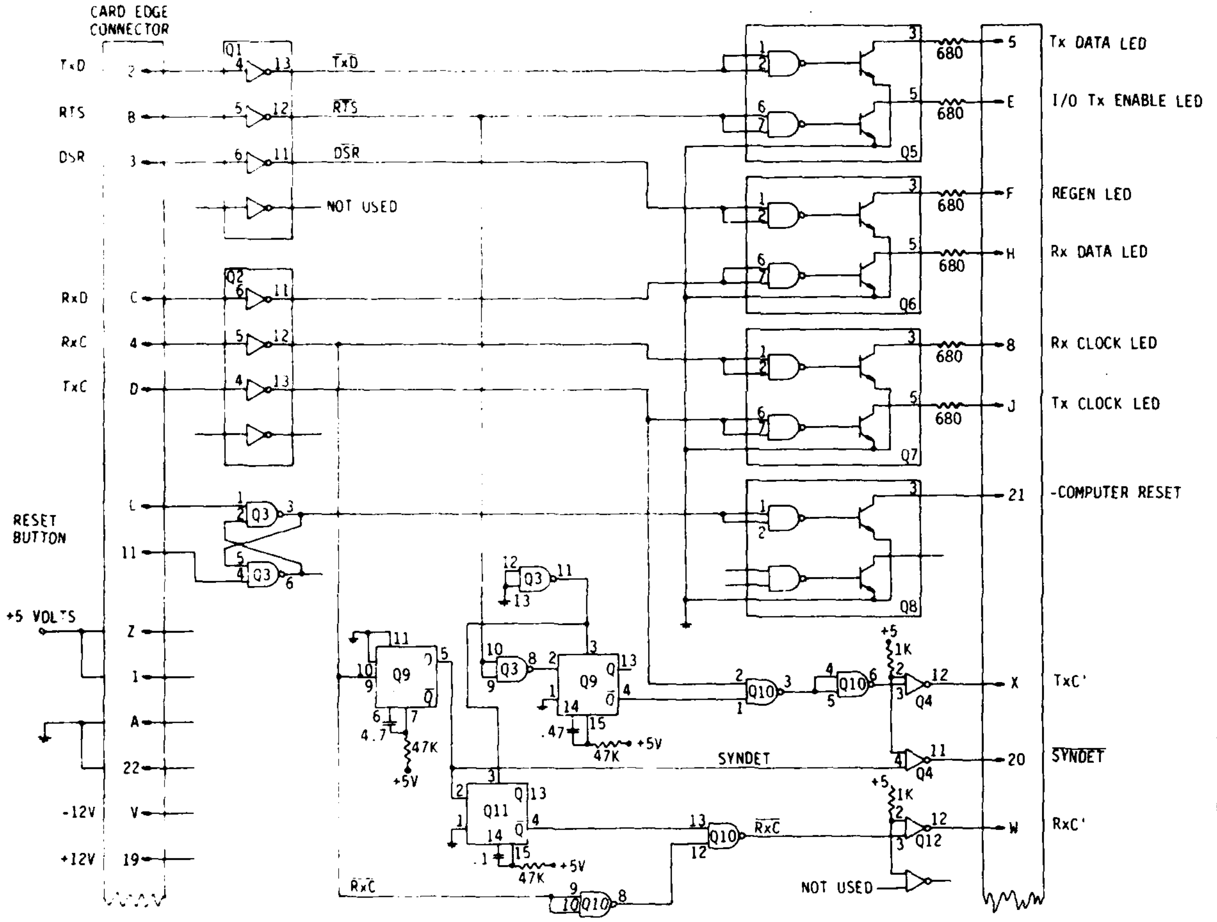


Figure 3-3b. Interface Schematic, Sheet 6



Q1	75154
Q2	75154
Q3, Q10	7400
Q4, Q12	75150
Q5, 6, 7, 8	75461
Q9, Q11	74123

Figure 3-4. Computer-to-AFSATCOM I/O Interface Schematic

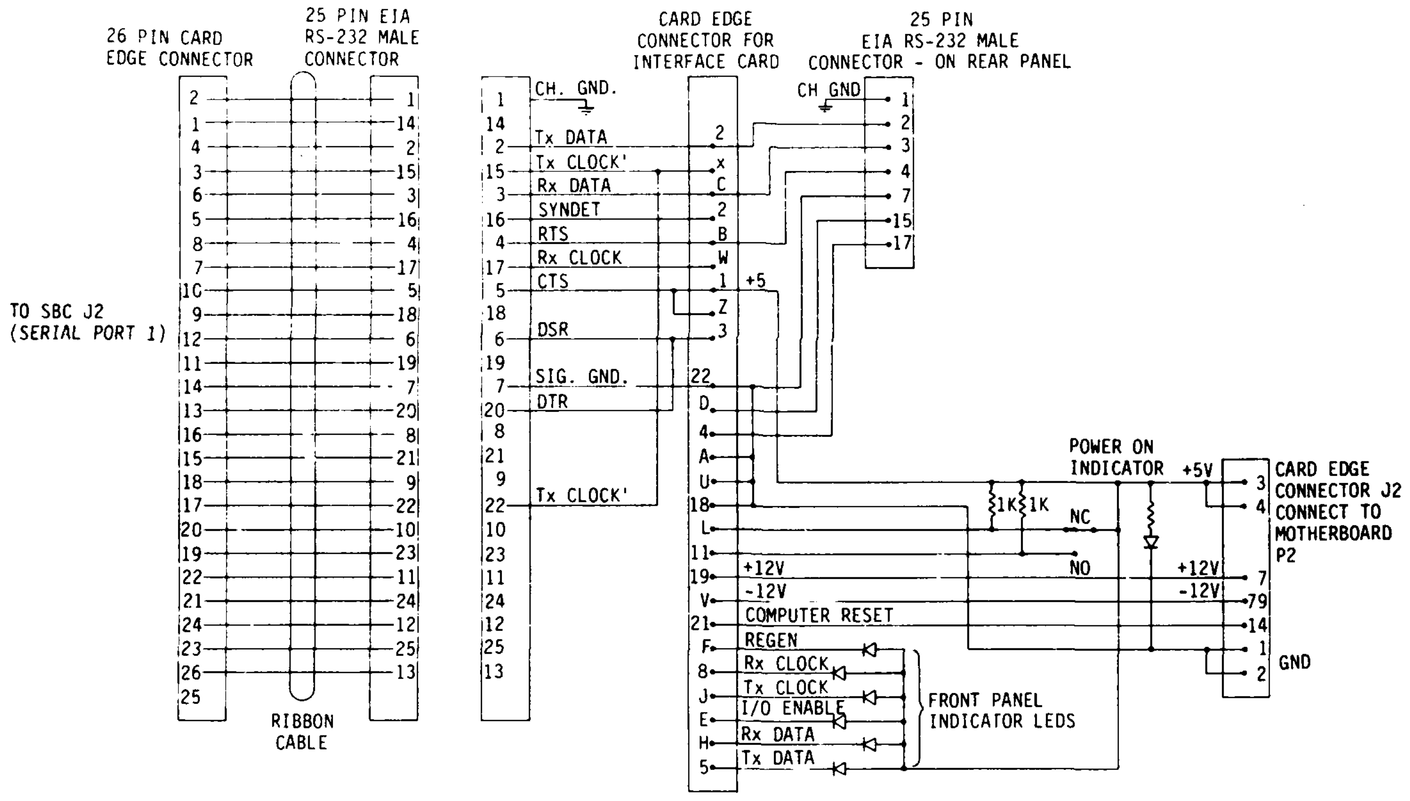

26 PIN CARD EDGE CONNECTOR
25 PIN EIA RS-232 MALE CONNECTOR
TO SBC J2 (SERIAL PORT 1)
RIBBON CABLE
CARD EDGE CONNECTOR FOR INTERFACE CARD
25 PIN EIA RS-232 MALE CONNECTOR - ON REAR PANEL
CH. GND.
Tx DATA
Tx CLOCK'
Rx DATA
SYNDET
RTS
Rx CLOCK
CTS
DSR
SIG. GND.
DTR
Tx CLOCK'
CH GND
+5
POWER ON INDICATOR
+5V
1K
1K
NC
NO
+12V
-12V
COMPUTER RESET
REGEN
Rx CLOCK
Tx CLOCK
I/O ENABLE
Rx DATA
Tx DATA
FRONT PANEL INDICATOR LEDS
CARD EDGE CONNECTOR J2 CONNECT TO MOTHERBOARD P2
GND

Figure 3-5. Internal Wiring Harness Connecting Computer, AFSATCOM Interface, and Front Panel

The USART chips used in the single board computer have a request-to-send (RTS) output which serves the same function as I/O TX enable does in the AFSATCOM system. Provision was made on the board to use separate external (to the computer) bit clocks for TX and RX data. In the synchronous mode, the USART shifts out one data bit for each rising edge of the clock signal. The AFSATCOM ASR also shifts out one data bit on each rising edge of the clock. This suggests that interfacing the computer serial I/O port to the AFSATCOM modem is as convenient as connecting an ASR to the modem.

One property of the Intel USART chip, not mentioned in the Intel literature, is that after being reset, the USART ignores the first clock pulse. That is, the first bit of the message is not shifted out until the rising edge of the second clock bit. This causes a bit slip at the very beginning of the message transmission, and causes the receiving terminal to print a message which is totally garbled.

This problem was corrected by building the circuit (see schematic, figure 3-4) consisting of 1/2 of dual one-shot Q9, two gates from Q10, and one gate from Q3. This circuit takes the $\overline{\text{TXC}}$ (transmit clock inverted) from the modem and the $\overline{\text{RTS}}$ (request to send inverted) signal from the computer and generates the signal called TXC′, which is used as the USART TX clock. TXC′ is identical to TXC except that TXC′ has an extra pulse at the beginning. When the I/O TX enable line goes high, the modem sends a 36 bit preamble before starting TXC, giving more than adequate time to insert one extra clock pulse. The timing diagram is shown in figure 3-6.

3.2.2 Receive Data Timing

When receiving, the USART requires a voltage transition on pin 16 to signal it to begin assembling the incoming bits into characters. The signal which serves this function is called SYNDET, and is generated from the receive clock (RXC) by a retriggerable one-shot (1/2 of Q9 on the schematic, figure 3-4). This one-shot is adjusted to have a period equal to about 40 clock periods. It provides a signal which transitions to a logic "1" when the received bit clock starts and remains in the "1" state until after the clock has stopped. The transition from "0" to "1" is used to establish character synch, and the transition from "1" to "0" is used to indicate the end of a received message. The message controller defines the end of message to have occurred when the bit clock has stopped for about 40 bits (5 characters).

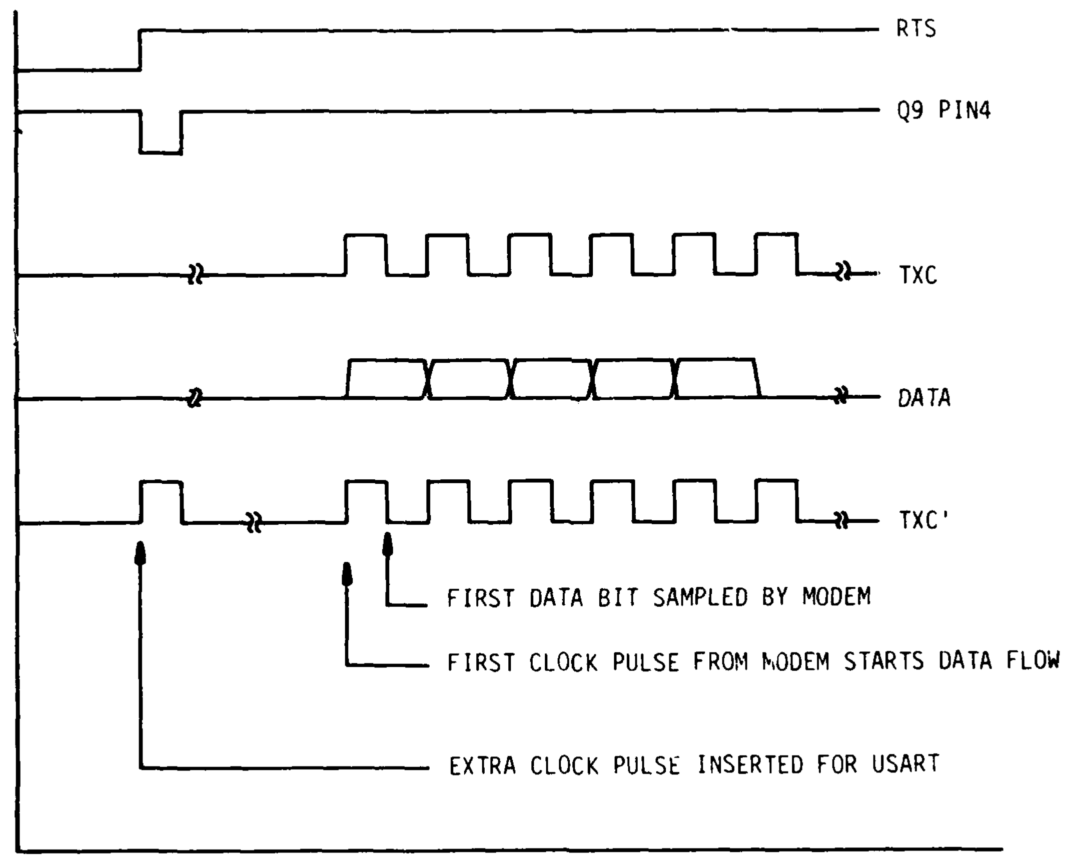

RTS
Q9 PIN4
TXC
DATA
TXC'
FIRST DATA BIT SAMPLED BY MODEM
FIRST CLOCK PULSE FROM MODEM STARTS DATA FLOW
EXTRA CLOCK PULSE INSERTED FOR USART

Figure 3-6. Timing Diagram for Transmit Data

The rising edge of the SYNDET pulse also triggers another one-shot (1/2 of Q11 on the schematic). This one-shot produces a pulse about 1 ms wide. The pulse is used to delay the start of the first RX clock pulse by that amount to insure that SYNDET rises and stabilizes before the first RX clock pulse rises. The RX clock signal modified in this manner is referred to as RXC′. A timing diagram for the receive data and associated clock pulses is shown in figure 3-7.

Prior to the use of the above solution, the first RX clock pulse was not delayed and a situation known as a "critical race" occurred. When SYNDET won the race, the message was printed properly. When the first clock pulse won the race, the message was garbled. Each condition occurred about 50% of the time.

All messages have been received properly without bit slips as a result of delaying the rising edge of the first clock period.

3.3.3 Signal Level Compatibility

The SBC-544 computer I/O lines are designed to be compatible with EIA standard RS-232C. This standard states simply that the mark and space levels will be ≤ -3 V and $\geq +3$ V, respectively. The line drivers used by Intel are standard 1488 chips, delivering about ± 12 V. The 1489 line receivers used by Intel make their mark/space decision at -3 V and +3 V as required by RS-232C, but they are undamaged by voltages up to ± 30 V. The modem was built to MIL-D-188C standards which state that the mark and space levels shall be $+6 \pm 1$ V and -6 ± 1 V, respectively. Since the 1489 line receivers can withstand up to ± 30 V, the signals from the modem can be applied directly to the computer board. However, the ± 12 V from the computer RTS and TX data lines must be limited to ± 6 V. Fortunately, the 1488 RS-232 line drivers are designed to be used as level convertors simply by using diodes to clamp the output voltage at the desired level. In fact, the output of the 1488 driver can be connected to any voltage between +25 V and -25 V without damage.

In this case, zener diodes connected back-to-back were used to clamp the RTS and TX data lines at ± 6 V so that the AFSATCOM modem requirements are met.

The only other compatibility problem was that RS-232 defines a mark as the low state, and MIL-D-188C defines a mark as a high. This was easily taken care of with software.

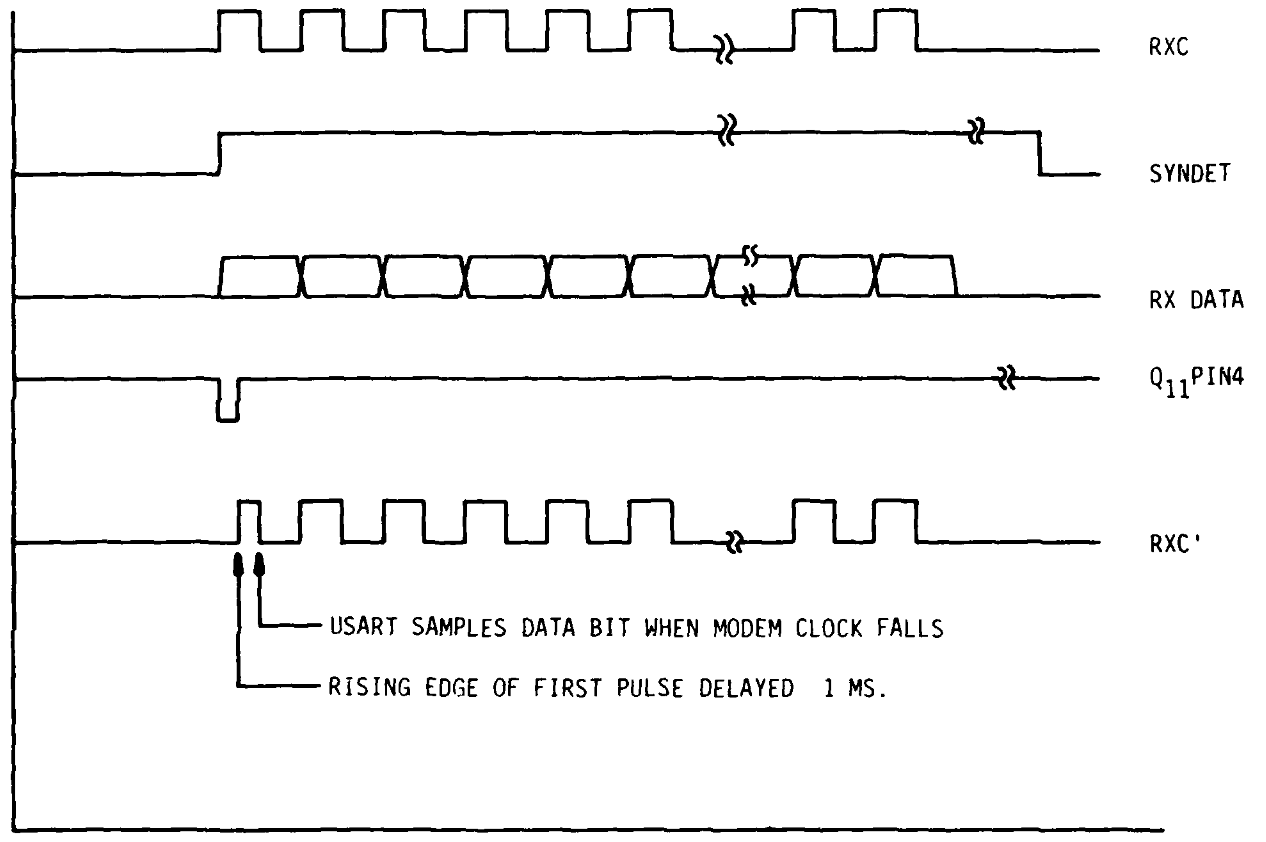

RXC
SYNDET
RX DATA
Q_{11}PIN4
RXC'
USART SAMPLES DATA BIT WHEN MODEM CLOCK FALLS
RISING EDGE OF FIRST PULSE DELAYED 1 MS.

Figure 3-7. Timing Diagram for Receive Data

3.3.4 Reset Circuit

A pair of NAND gates (part of Q3) was cross-connected to form a flip-flop and used to debounce the push button switch on the front panel to provide a clean reset signal.

3.3.5 Indicator Lamps

Q5, Q6, and Q7 are used as lamp drivers to illuminate six light emitting diodes (LEDs) on the front panel. The indicators are used to show:

1. I/O TX enable line status
2. TX clock status
3. TX data bit status
4. RX clock status
5. RX data status
6. Regenerative or non-regenerative test

The I/O TX enable indicator will glow brightly when the transmitter is on. The TX clock and RX clock indicators will glow at half brightness when their respective clocks are running, because the clock, being a square wave, is on only 50% of the time. The TX data and RX data indicators blink as data is being transferred. The REGEN lamp indicates the status of the DSR bit from the computer. DSR is an RS-232 signal not used by the dual modem that is used here to provide a visual indication of the type of test. This signal is also sampled by the DTR line to provide a 1-bit memory for the computer to use in order to "remember" what type of test is being performed.

SECTION 4

SOFTWARE DESCRIPTION

The message controller software consists of an executive routine and 17 subroutines. The entire program code resides in ROM locations 0000 through 0309. Nine tables (TA1 through TA9), which are used by the program, reside in ROM locations 03B6 through 0572 (see ROM map in figure 4-1a).

Note that the routines REGEN and NONREGEN listed in figure 4-1a are not subroutines. Although both started out as subroutines, they are now entered and exited by jump instructions and have become part of the executive routine.

The program is designed to begin execution at location 0000, so that the power-up reset circuitry on the SBC-544 board will cause the program to begin running as soon as the power to the message controller is turned on. No other action is required from the operator to start the program running.

Once running, the executive routine initializes the on-board timer and I/O chips and partitions the random access memory (see RAM map in figure 4-1b). The upper 16 bytes of RAM are used as scratch pad memory. The space immediately below the scratch pad is reserved for the stack. The lowest RAM address (8000) is the beginning of the buffer used for storage of the test message. Additionally, when the ASR emulation mode is being used, the software further partitions the memory to create space for received messages.

Since the scratch pad is separated from the beginning of the test message buffer by over 16,000 bytes, there is no need for any buffer management in the program. Even when using the ASR emulation mode, the chance of any message filling the allotted space completely is nil.

The ASR emulation mode utilizes the program ASRSIM, located in memory addresses 0376 to 03B5. ASRSIM is itself an executive routine, which is invoked by typing "A" in response to the prompt: "REPETITIVE MESSAGE TEST, ASR EMULATION, OR SPECIAL TEST? TYPE R/A/S." ASRSIM uses the RCVMSG subroutine (locations 0340 through 0375) and table TA10, as well as a number of the 17 subroutines used by the main program.

Four utility subroutines were provided to facilitate future expansion. These are PORT2OUT, PORT3OUT, CHARIN2, and CHARIN3

	Start	End	Routine	Function
	0000	011C	EXECUTIVE ROUTINE	
1	011D	0149	XMTMSG	Transmits a message via AFSATCOM Recognizes certain control functions
2	014A	015C	XMTON	Turns AFSATCOM transmitter ON
3	015D	0181	XMTOFF	Turns AFSATCOM transmitter OFF
4	0182	018D	CONIN	Inputs one character from keyboard
5	018E	01A9	TOGGLE	Inserts random data between messages for a user specified number of seconds
6	01AA	01B4	XMTMSG2	Transmits a table from memory via AFSATCOM
7	01B5	01C1	OUTMOD	Outputs one character to AFSATCOM modem
8	01C2	01E0	MSGNMBR	Transmits message number when SUB is encountered in message
9	01E1	01E8	ASKEY	Converts 4 LSBs of packed BCD to ODD parity ASCII
10	01E9	020C	PAUSE	Inserts a user-specified delay between messages
11	020D	0214	EPARITY	Outputs next character of message with even parity when US is encountered in message
	0215	021B	REGEN	Sets USART 1 DTR bit = 1 when regenerative test is specified
	021C	0222	NON-REGEN	Sets USART 1 DTR bit = 0 when non-regenerative test is specified
12	0223	0238	PRTMSG	Outputs a table from memory to printer
13	0239	0266	TOGGLE2	Inserts user specified number of seconds of random data in test message when FS is encountered
14	0267	0294	MSGIN	Inputs a message from keyboard to memory
15	0295	02FF	MSGCOUNT/DELAYIN	Inputs number of test messages to be sent and delay between messages
16	02F7	0301	BCDIN	Inputs one character; converts to BCD if between 0-9
17	0302	0309	CLEARCOUNT	Clears message count in locations BFF2-BFF3
18	030A	031A	PORT2OUT	Outputs a table via USART 2
18	031B	032B	PORT3OUT	Outputs a table via USART 3
19	032C	0335	CHARIN2	Inputs one character via USART 2
19	0336	033F	CHARIN3	Inputs one character via USART 3
20	0340	0375	RCVMSG	Inputs, prints, and stores message from AFSATCOM modem
	0376	03B5	ASRSIM PROGRAM	Makes computer and console partially emulate an AFSATCOM ASR
	03B6	0577	OPERATOR PROMPT TABLES	TA1 through TA10

Figure 4-1a. ROM Map, Test Message Generator Controller Program

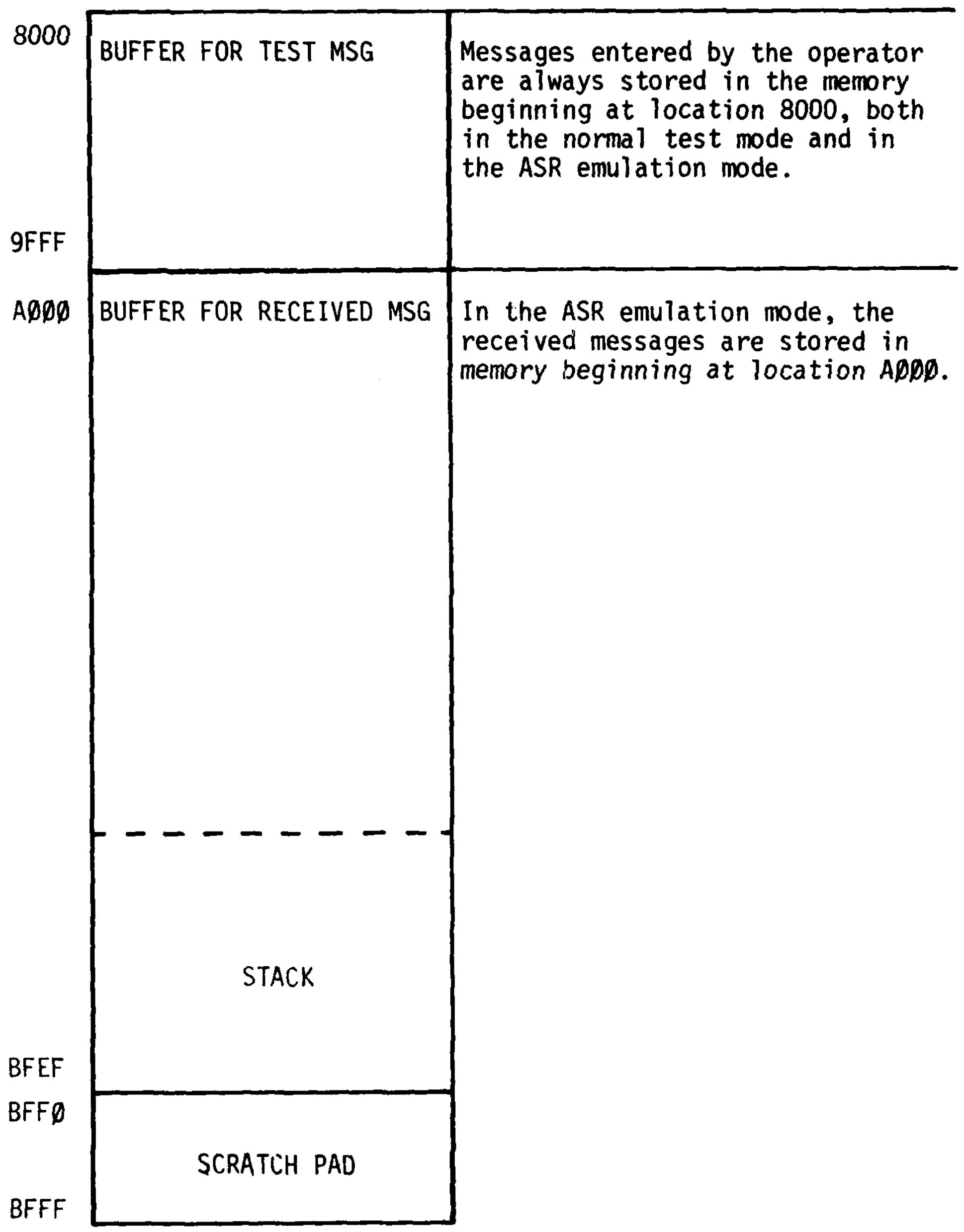

8000
BUFFER FOR TEST MSG
Messages entered by the operator are always stored in the memory beginning at location 8000, both in the normal test mode and in the ASR emulation mode.
9FFF
A000
BUFFER FOR RECEIVED MSG
In the ASR emulation mode, the received messages are stored in memory beginning at location A000.
STACK
BFEF
BFF0
SCRATCH PAD
BFFF

Figure 4-1b. RAM Map, Test Message Generator Controller Program

(locations 030A to 033F). None of these four subroutines is currently used. These subroutines support I/O on the single board computer serial I/O ports 2 and 3, which are not used at the present time.

The most likely use of these four subroutines would be to call them from another program which would begin execution at location 0800. Memory locations 0800 through OFFF are the addresses assigned to the socket labeled "PROM 1" on the single board computer. This socket, which can accept a 2716 ROM chip, is currently empty and is reserved for future expansion. Program execution can be made to jump to location 0800 by typing "S" in response to the prompt "REPETITIVE MESSAGE TEST, ASR EMULATION, OR SPECIAL TEST? TYPE R/A/S". Thus, capability to add any additional program for any need which may develop is built in.

Some examples of tasks that could be accomplished by a supplemental program at location 0800 might include the following:

- Receiving messages, analyzing them for errors, printing the messages as received on the main console, and printing reduced data on another printer plugged into one of the spare I/O ports.
- Receiving messages, looking for a header, and printing messages with a header on one printer and messages without a header on another printer.
- Receiving messages, looking for a header, and initiating the transmission of a message or a sequence of messages in response to the header.
- Receiving and printing messages and putting them on a storage device either as-received or in processed form. The storage device would be connected to one of the spare RS-232C I/O ports.
- Appending a date-time sequence to each received message. A system clock connected to one of the spare ports would enable the processor to perform this function.

The above list is not all-inclusive but serves to illustrate the kinds of tests which could be supported with the addition of only a single programmed 2716 ROM chip.

4.1 EXECUTIVE ROUTINE

Immediately upon power-up, the 8085 microprocessor begins executing instructions at location 000H. The assembly language instruction ORG 000H tells the assembler to begin assembling machine code at location 000H (see appendix A, sheet 1).

4.1.1 Initializing the Single Board Computer

The power-up reset initializes not only the 8085 chip, but all of the I/O chips, timer chips, and interrupt circuits as well. However, it should be noted that the serial I/O ports are not initialized directly by the power-up reset pulse. Instead, the reset pulse resets the parallel I/O chip, and one of the parallel output lines is used to reset the serial I/O (USART) chips. This procedure allows the program to reset all of the USART chips quickly and easily. However, it also means that the reset must be deliberately removed from the USARTs after a reset and prior to attempting the initialization of the USART chips. Lines 7 through 10 of the assembly language program (appendix A, sheet 1) remove the reset from the USART chips.

Lines 16 through 31 set the baud timers for ports 0, 2, and 3 to provide a 19.2 kHz clock to the USART chips. The 19.2 kHz is 64 times the serial data transfer rate of 300 baud. In the asynchronous mode, the USART requires the clock to be either 16 or 64 times the baud rate, with 64 times giving the best error performance.

No on-board baud timer is provided for USART 1, as it operates with the AFSATCOM modem in the synchronous mode and the clock is provided by the modem. For timer programming details, refer to the SBC-544 Instruction Manual, pages 3-7 through 3-11.

Lines 42 through 66 of the program (appendix A, sheet 2) initialize the USARTs. Ports 0, 2, and 3 are set to the asynchronous mode, with a data transfer rate of 300 baud. Port 1 is initialized to the synchronous mode. The 300 baud rate was chosen for port 1 because it is the fastest that the TI-765 printer can operate in the non-buffered mode. Since this rate is approximately three times the character rate of the AFSATCOM system, there is really no need to operate faster than 300 baud. The choice of baud rate for ports 2 and 3 was totally arbitrary, and can be changed easily if required. This could be done either by having the expansion program re-initialize the timers or by changing the existing program.

When the system operates in the synchronous mode, the USART chip requires that a synch character be programmed. The synch character can be any 8-bit or 16-bit sequence, and must be programmed into the USART during the initialization process. The dollar sign was chosen as the synch character for the message controller because it is printable and seldom, if ever, used in normal AFSATCOM communications. If the transmit buffer is empty, the USART chip inserts synch characters in the data stream to keep up a continuous flow of data. A printing character was chosen for the synch character so that test personnel would be aware of any extra characters inserted into a test message.

4.1.2 Selecting the Mode of Operation

The flow charts of the executive routine in figures 4-2a, 4-2b, 4-2c, and 4-2d explain sheets 3, 4, and 5 of the assembly language program (appendix A).

The software can print any table in memory on the system console by loading the address of the first character in the table into the HL register pair and calling the PRTMSG subroutine. The prompt message "REPETITIVE MESSAGE TEST, ASR EMULATION, OR SPECIAL TEST? TYPE R/A/S" is stored in table TA1. Printing this prompt is accomplished with two instructions: LXI H,TA1 and CALL PRTMSG. These instructions appear in lines 72 and 73 of the listing in appendix A. The subroutine PRTMSG controls the details of actually printing the prompt message. PRTMSG is described section 4.13.

After the CONIN subroutine inputs the operator's response, PRTMSG is used again in line 78 to output a short message consisting of carriage return and line feed. The carriage return and line feed characters are stored in table TA9.

Following the carriage return/line feed, the character typed by the operator is tested. If it is an upper case "S", for special test, the program execution jumps to location 0800, which has been reserved for expansion. If an upper case "A" is typed, for ASR emulation, program execution jumps to location 0376, represented by the symbolic address ASRSIM. If an upper case "R" is typed, program execution continues with the next instruction. If the typed character is neither an "R", an "S", nor an "A", the initial prompt is repeated and the operator must respond with one of the three allowable choices. Typed character testing is done in lines 80 through 86 of the program listing, appendix A, sheet 3.

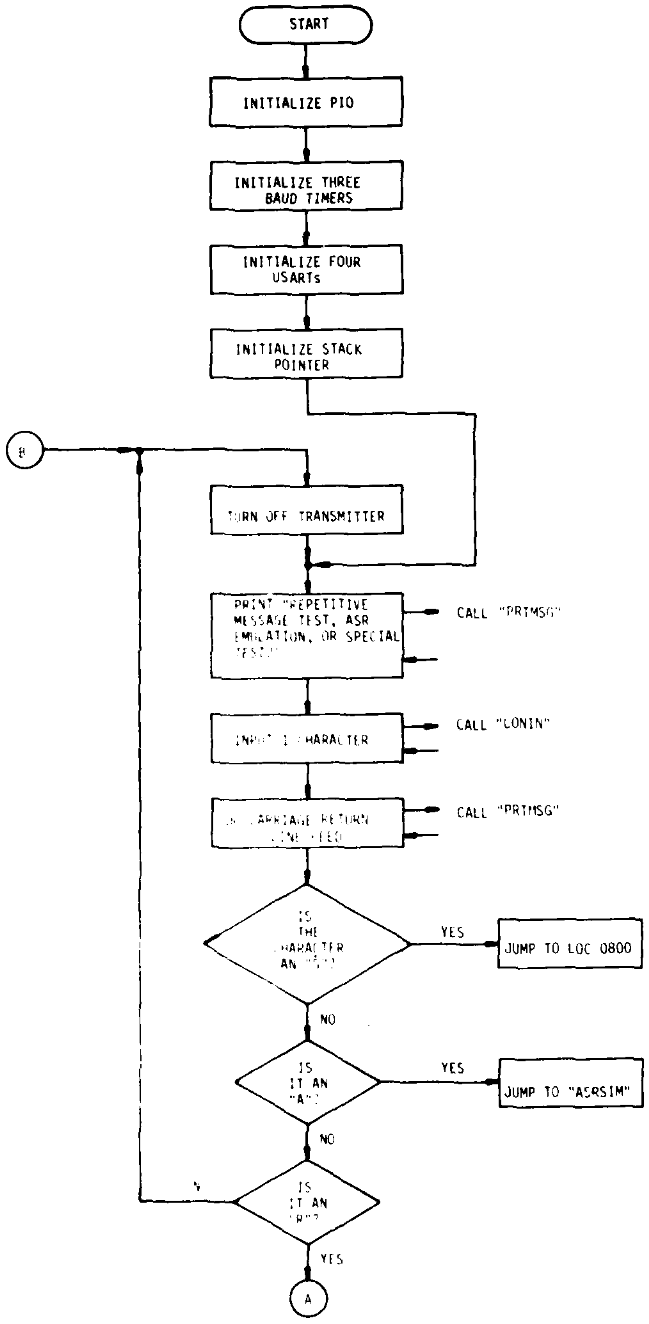


Figure 4-2a. Main Program Flowchart A

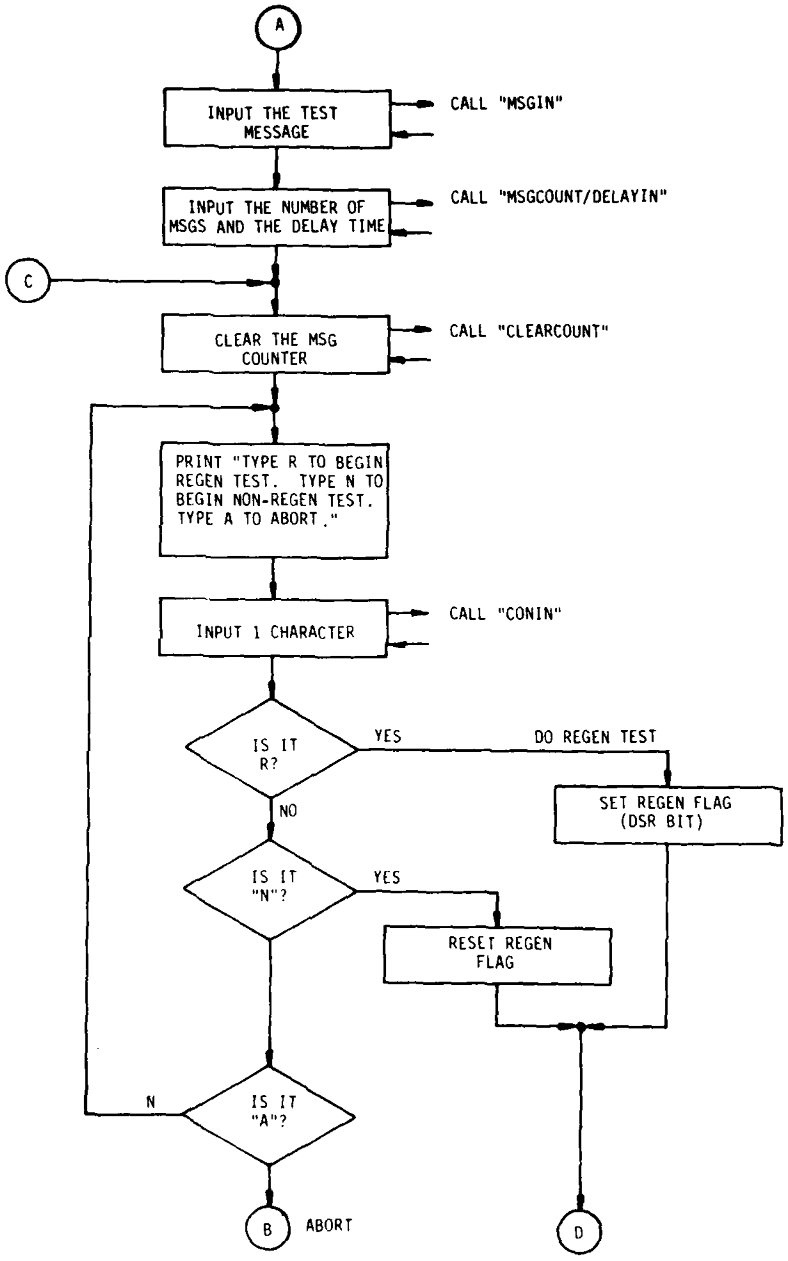


Figure 4-2b. Main Program Flowchart B

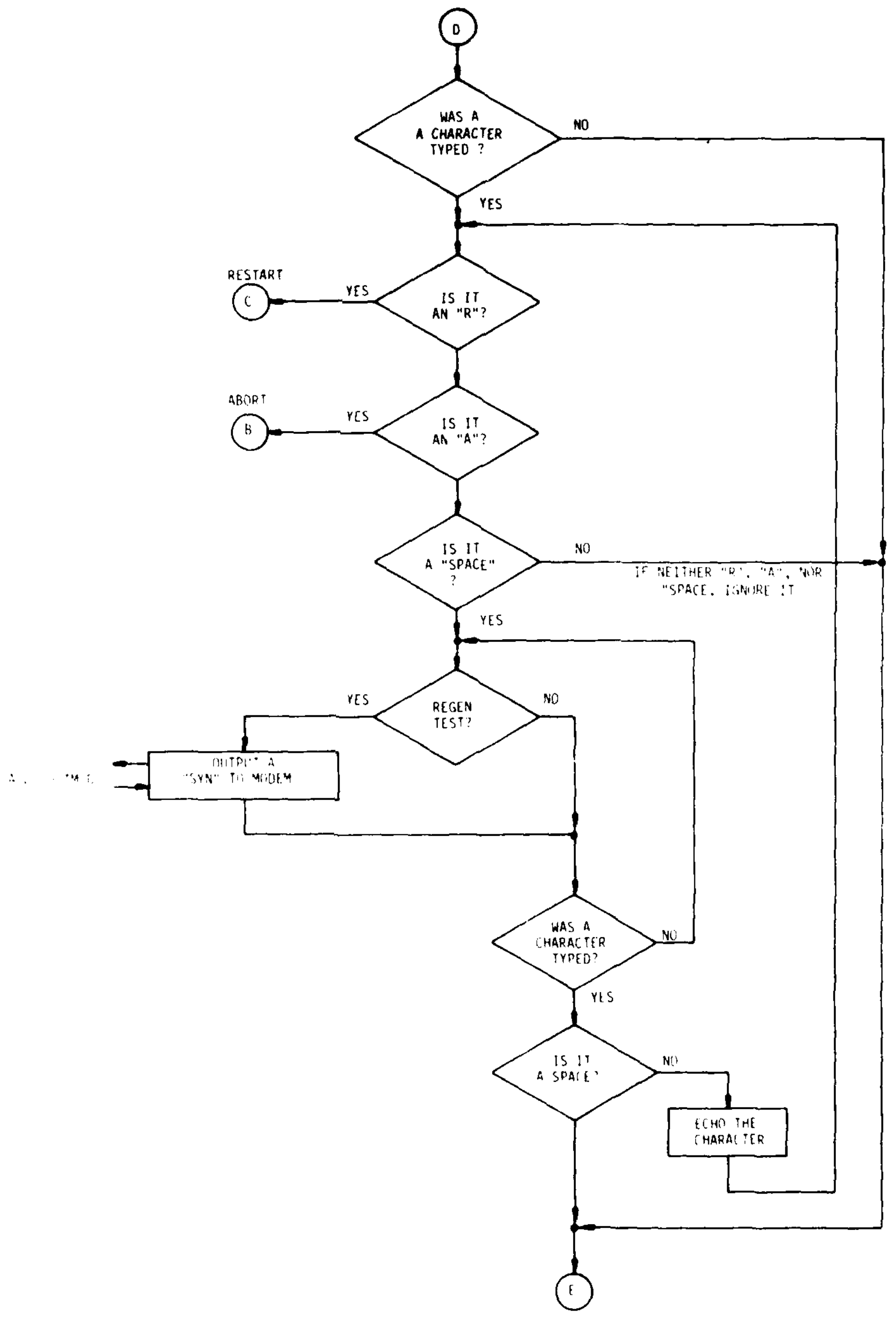


Figure 4-2c. Main Program Flowchart C

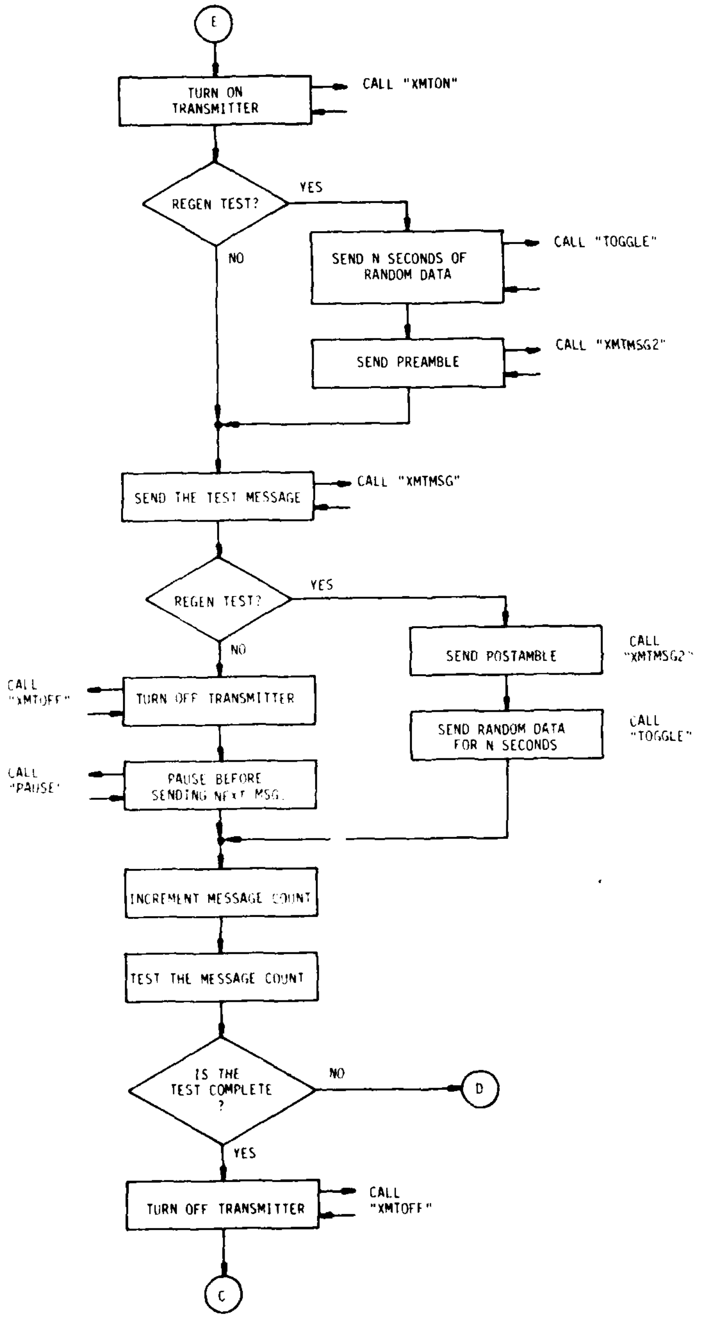


Figure 4-2d. Main Program Flowchart D

4.1.3 Entering Test Parameters

If an "R" is typed, execution of the main program continues. The operator inputs the test message with a subroutine called MSGIN, which is described in section 4.2.14.

The operator next uses a subroutine called MSGCOUNT/DELAYIN to enter two parameters: the number of times the message is to be transmitted, and the number of seconds of delay between messages. MSGCOUNT/DELAYIN is described in section 4.2.15.

After entering the test parameters, the software clears the internal count of the number of messages sent by calling the CLEARCOUNT subroutine (section 4.2.17).

4.1.4 Executing the Test

The test is now ready to begin. The instructions in lines 92 and 93 (appendix A, sheet 3), cause table TA2 to be printed: "TYPE R TO BEGIN REGEN TEST. TYPE N TO BEGIN NON-REGEN TEST. TYPE A TO ABORT." Typing "A" causes program execution to jump back to the point just after initialization where the operator is asked what kind of test is being run. Typing "R" or "N" causes the REGEN flag to be set or reset, followed by the transmission of test messages. The REGEN flag is set by setting the DTR status line on the USART to a logic "1". Since the DTR output is connected to the DSR input, the USART can test the status of its own DTR line by reading the DSR line. (See SBC-544 Instruction Manual for a description of the 8251A USART.) Besides giving a 1-bit memory which can be used to remember which type of test is being run, this technique provides external access to the status of this bit. Such access would not be possible if a memory location were used to store the flag. A lamp driver connected to the DTR line uses this feature to provide a visual indication of the type of test being run.

After the flag is set, the program checks to see if a character was typed. The first time the program arrives at this point, no character will have been typed, since the operator began the test by typing one character just a few microseconds earlier and could not possibly have had time to type another. However, program execution branches back to this point (D on the flow chart, figure 4-2b) each time the transmission of a test message has been completed. Testing the keyboard prior to transmission of each test message provides the operator with an opportunity to abort, restart, or temporarily suspend testing.

If no character was t,ped prior to the program branching back to point D, execution branches to point E on the flow chart, and transmission of a message is initiated. If a character was typed, it is tested. If the typed character is "A", the test is aborted and the operator will be asked if he desires a repetitive message test, an ASR emulation, or a special test.

If an "R" (for RESTART) is typed prior to point D in the main program, the test will be restarted. That is, the message count will be cleared and the operator will be prompted to begin a regenerative or a non-regenerative test. Typing "R" allows the operator to start the test over without having to re-enter the test message and the test parameters.

If a space has been typed prior to point D, the test is suspended until another character is typed. In the case of a non-regenerative test, the processor simply waits until another character is typed. Suspending a regenerative test by typing a space causes synchronous idle (SYN) characters to be sent until another character is typed.

Once the test has been suspended by typing a "space", the test can be aborted by typing "A", restarted by typing "R", or can be made to resume by typing any character except "R" or "A". Testing the typed character for "R", "A", or "space" occurs in lines 105 through 128 of the program listing. If, prior to point D, a character other than "R", "A", or "space" has been typed, it will be ignored.

The actual transmission of a test message begins at point "E" on the flow chart (figure 4-2d). The first step in sending a message is to turn on the transmitter. Calling the XMTON subroutine does this automatically. During regenerative testing, the transmitter will usually be on at this point, but this is of no consequence. Turning on the transmitter when it is already on leaves everything unchanged.

When the test is not a simulation of a regenerative channel, the message can be sent as soon as the transmitter is turned on. The main program sends the message by pointing the HL register pair to the address of the first character of the message in the memory buffer and calling the subroutine XMTMSG.

When the test is a simulation of a regenerative channel, the main program must send the toggle table followed by a preamble before the test message can be transmitted. The toggle table is sent by the subroutine TOGGLE, and the preamble is sent by XMTMSG2. The difference between XMTMSG and XMTMSG2 is described in section

4.2.6. Lines 129 through 138 of the assembly language listing represent the portion of the program that turns on the transmitter and sends the test message (appendix A, sheets 4 and 5).

After the test message has been transmitted, if the test is non-regenerative, the transmitter shuts off and the program pauses for the number of seconds between messages entered by the operator during the entry of test parameters.

In the regenerative simulation mode, after the message has been sent, the computer sends the postamble from table TA4 and then sends random data for the number of seconds delay between messages the operator has entered.

After the pause or toggling, the message count stored in memory locations BFF2 and BFF3 is incremented and tested. If the desired number of messages has been sent, the transmitter turns off and program execution jumps to point C on the flow chart in figure 4-2d. If the test was non-regenerative, the transmitter will already be off when the test of the message count shows that the test is complete. Turning off the transmitter when it is already off does not cause a problem.

If the test of the message count shows that the test is incomplete, the program branches to point "D", where the keyboard is tested for typed characters prior to sending the next message.

4.2 SUBROUTINES

The subroutines of the executive program are described in the following paragraphs. See the ROM map in figure 4-1a for a summary of subroutine functions.

4.2.1 XMTMSG

A flow chart of XMTMSG is shown in figure 4-3. The assembly language listing is given in appendix A, sheet 6.

Before this subroutine is called, the calling routine must load the HL register pair with the address of the first byte of the test message. XMTMSG fetches each byte of the test message in turn, tests it, and outputs it to the AFSATCOM modem if it is not one of five special characters.

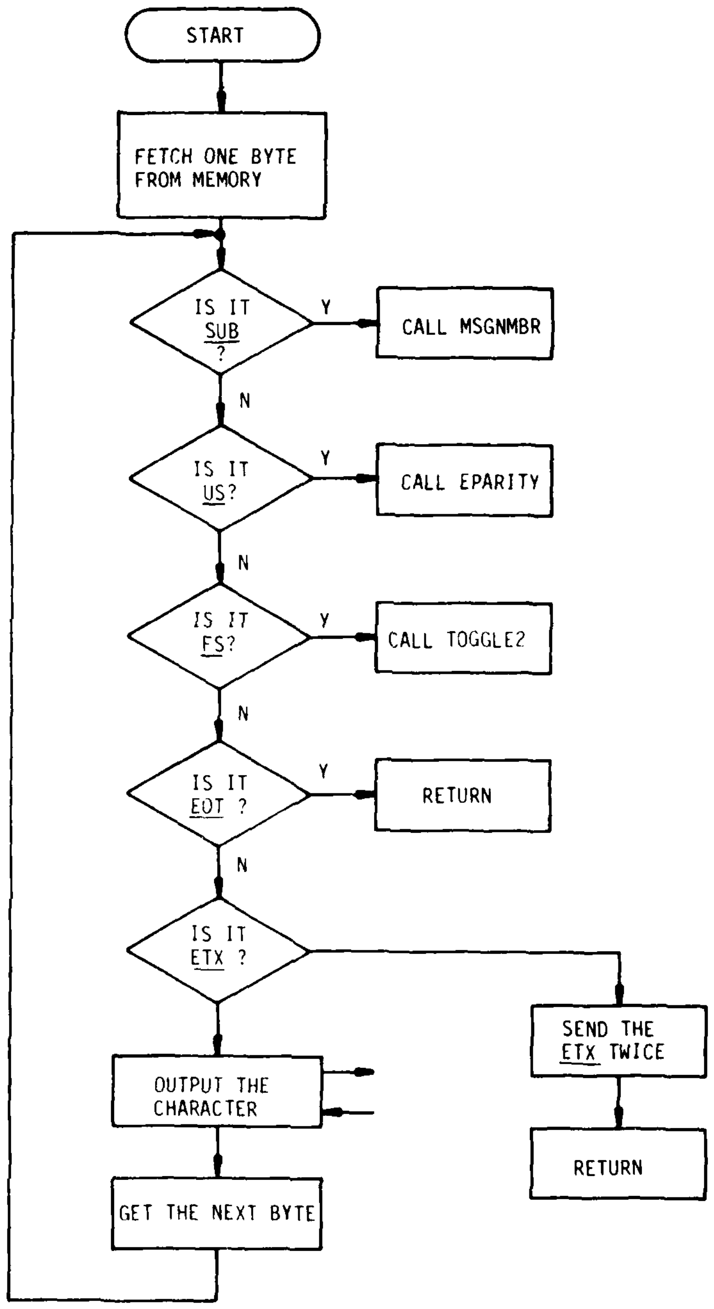


Figure 4-3. XMTMSG Subroutine

If the character is SUB (control Z), US (control /), or FS (control ","), then one of the subroutines MSGNMBER, EPARITY, or TOGGLE2 will be called. If the character is EOT (control D), transmission will terminate. If the character is ETX (control C), ETX will be sent twice and message transmission will terminate.

4.2.2 XMTON

A flow chart of XMTON is shown in figure 4-4, and the assembly language listing appears in appendix A, sheet 7. This subroutine raises the I/O transmit enable line from the message controller from -6 V to +6 V and turns on the transmitter.

To turn on the transmitter during a non-regenerative test, XMTON loads the control word 35H into the control register of USART 1. Control word 35H sets the RTS, error reset, TX Enable, and RX Enable bits. For a regenerative test, control word 37H is used to set all of the above bits plus the DTR bit.

Operation of the 8251A USART is detailed in the SBC-544 Instruction Manual, sections 3-48 through 3-57.

4.2.3 XMTOFF

This subroutine, shown in the flow chart of figure 4-5, turns off the AFSATCOM transmitter after determining that the AFSATCOM modem has sampled the last bit of the message. In order to determine that the last bit has been sent, the program first tests the USART until the TX empty bit becomes true. TX empty indicates that the last bit in the USART has been put onto the serial output line. This occurs on the rising edge of the modem TX clock pulse. Since the bit is not sampled until the falling edge of the clock pulse, the data must be held on the output line for at least one-half of a TX clock period after the TX empty bit goes true. This need was accommodated by bringing the TX clock into the computer via one of the parallel I/O ports. This enables the TX clock to be tested. When the clock returns to the logic "0" state, the USART servicing serial port 1 is reset. Testing the status word and clock state, and resetting the USART occur in lines 238 through 246 of the program listing in appendix A, sheet 8.

Resetting the USART causes the RTS (I/O TX enable) to go low, turning off the transmitter. After reset, the USART must be re-initialized.

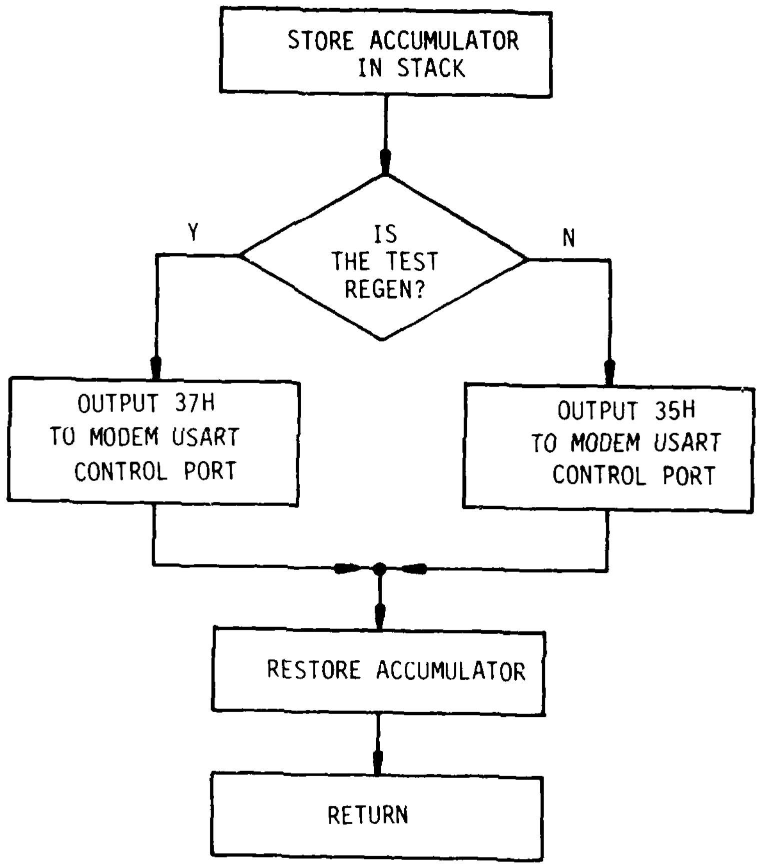

STORE ACCUMULATOR IN STACK
IS THE TEST REGEN?
Y
N
OUTPUT 37H TO MODEM USART CONTROL PORT
OUTPUT 35H TO MODEM USART CONTROL PORT
RESTORE ACCUMULATOR
RETURN

Figure 4-4. XMTON Subroutine

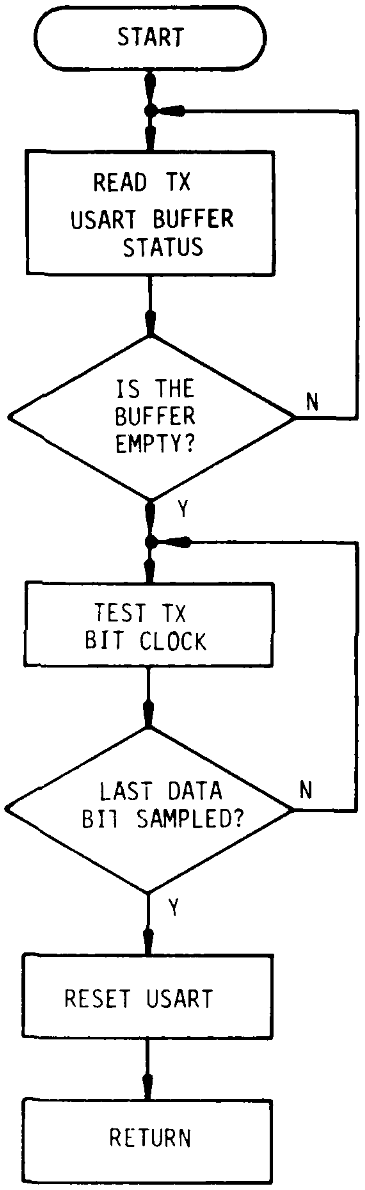


Figure 4-5. XMTOFF Subroutine

The transmitter could be turned off more easily by simply commanding the RTS line into the low state. However, not resetting the USART after each message would create two problems. First, as discussed in section 3.1, an extra TX clock pulse is required at the beginning of a message for the first message following a reset. To avoid increasing the complexity of the hardware interface to make it distinguish the first message from the others, each message is preceded by a reset.

The second problem is that in the synchronous mode, when the transmit buffer runs empty, the USART inserts a synch character in the data stream. Even if the extra clock pulse is added to only the first message, an extra synch character would be sent at the beginning of all messages except the first.

> NOTE: The subtle properties of the USART when it is operating in the synchronous mode are not well described in the Intel literature, and many hours were spent developing a simple system which would operate satisfactorily with the AFSATCOM system without garbled messages and without inserting any extra characters in the message. Anyone desiring to change the procedure for turning off the transmitter should do so very carefully.

4.2.4 CONIN

CONIN is a very simple subroutine which inputs one character from the system keyboard. The flow chart is shown in figure 4-6, and the listing is in appendix A, sheet 9.

CONIN keeps testing the status word of USART 0. When the status word shows that a character has been typed, the data register of USART 0 is read into the accumulator, then written to the output data register of USART 0 so that it will be printed on the system console. The typed character is then returned to the calling routine in the accumulator.

4.2.5 TOGGLE

This subroutine, listed in appendix A, sheet 10 and flow charted in figure 4-7, is used when the test calls for simulating the output of a satellite regenerative channel. At the end of each message, TOGGLE inserts a data stream which simulates the random toggling of an idle channel. Table TA5 contains 10 characters chosen randomly, followed by a 00H end-of-file marker. At the

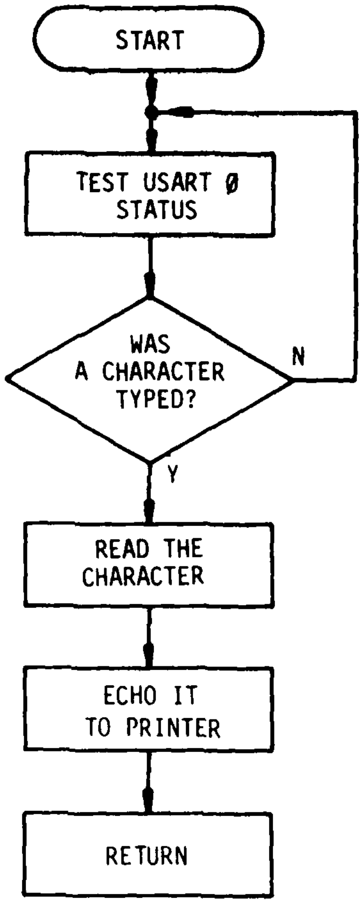


Figure 4-6. CONIN Subroutine

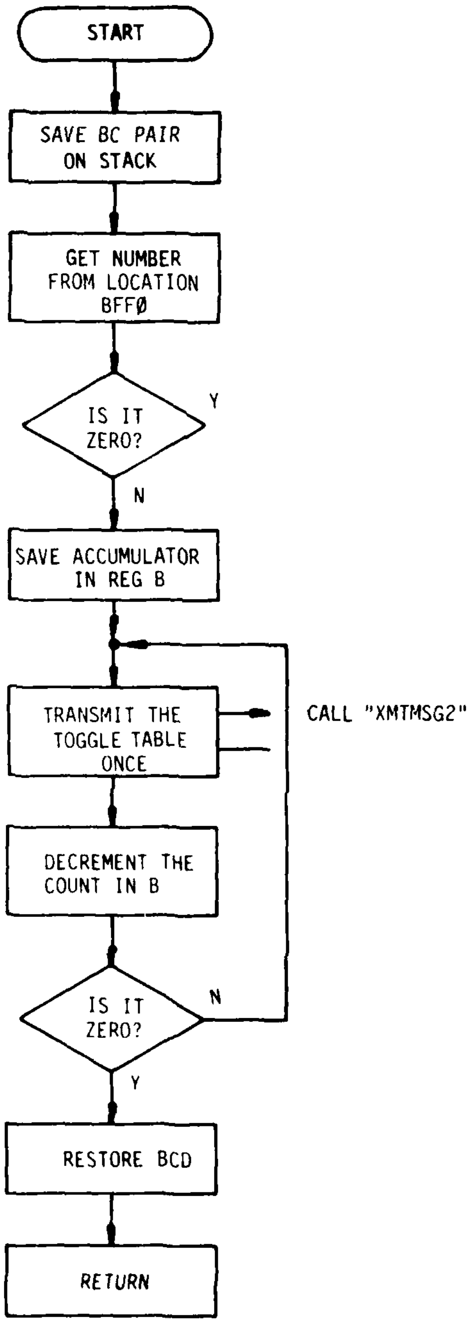


Figure 4-7. TOGGLE Subroutine

AFSATCOM rate of 75 baud, this 10 character (80 bit) table takes a little more than one second to transmit. Memory location BFF0 contains the number of seconds of delay between messages. TOGGLE uses this number to determine how many times to send table TA5. For instance, if the number stored at BFF0 is 07, table TA5 will be transmitted seven times before control is returned to the calling routine. The DELAYIN portion of MSGCOUNT/DELAYIN forces the operator to enter the number at location BFF0 before beginning the test. This number is stored in packed binary coded decimal (BCD) form.

4.2.6 XMTMSG2

XMTMSG2, shown in the flowchart of figure 4-8 and listed in appendix A, sheet 11, outputs a table from memory to the AFSATCOM modem. Prior to calling XMTMSG2, the calling routine must load the HL register pair with the address of the first character in the table to be transmitted. Each character is outputted in turn, exactly as stored in memory, until a 00H end-of-file marker is encountered. The 00H is not transmitted; it merely signals the end of file and causes the program execution to return to the calling routine.

XMTMSG2 differs from XMTMSG in that XMTMSG recognizes several control characters imbedded in the test message and acts on them, while XMTMSG2 does not. Another difference is that XMTMSG terminates on either an ETX or an EOT, while XMTMSG2 terminates on a 00H (even parity NUL character).

4.2.7 OUTMOD

This simple subroutine outputs the contents of the accumulator to the AFSATCOM modem. The flow chart is shown in figure 4-9 and the listing is in appendix A, sheet 12.

The accumulator, which contains the character to be transmitted, is stored temporarily on the stack while the USART transmit buffer is tested. When the USART is ready to accept data, the character is popped off the stack, complemented, and written into the USART transmit buffer. The complement is necessary because the computer is built to RS-232C standards and the modem is built to MIL-D-188C standards. One of the principal differences between the two standards is the reversal of the sense of the mark and space.

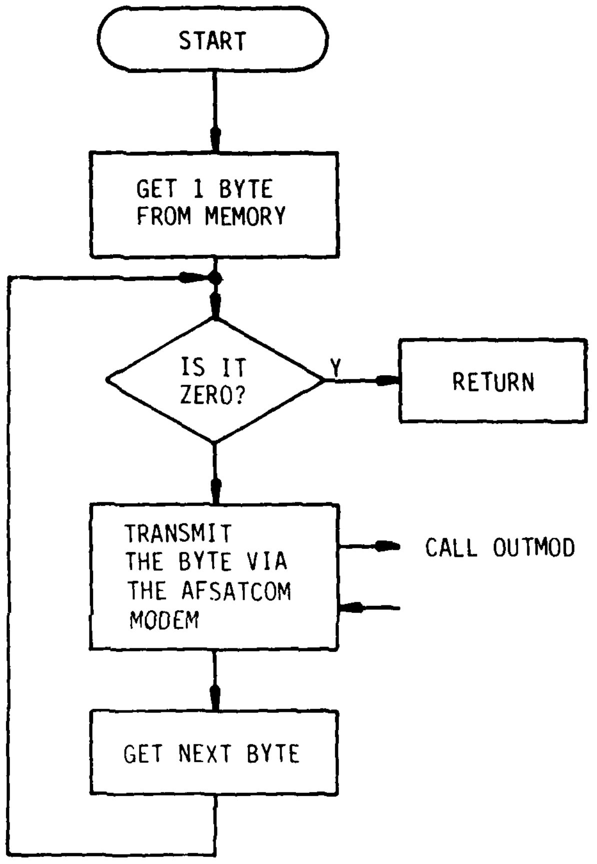


Figure 4-8. XMTMSG2 Subroutine

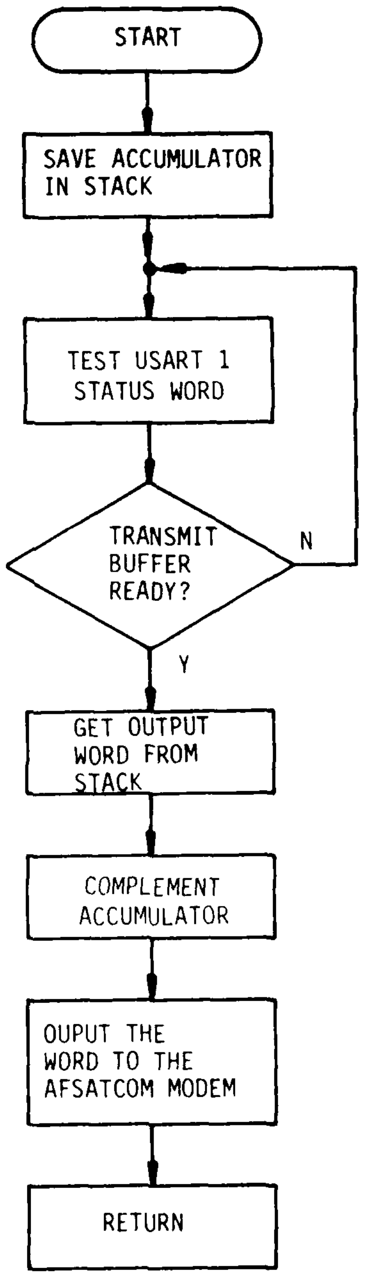

START
SAVE ACCUMULATOR IN STACK
TEST USART 1 STATUS WORD
TRANSMIT BUFFER READY?
N
Y
GET OUTPUT WORD FROM STACK
COMPLEMENT ACCUMULATOR
OUPUT THE WORD TO THE AFSATCOM MODEM
RETURN

Figure 4-9. OUTMOD Subroutine

4.2.8 MSGNMBR

MSGMNBR (see flowchart, figure 4-10, and listing, appendix A, sheet 13) is called by the XMTMSG subroutine when SUB (control Z) is encountered in the message. MSGNMBR takes the message count stored in memory locations BFF2 and BFF3 in packed BCD form, converts it to ASCII format, and transmits it via the AFSATCOM modem. The conversion to ASCII format is done by a subroutine called ASKEY.

Two things must be noted. First, the message count is a full four digit count, but only the three least significant digits are transmitted when MSGNMBR is called. Second, only two digits are transmitted to the modem by MSGNMBR. The least significant digit remains in the accumulator when the program returns to XMTMSG, which will output the last digit.

4.2.9 ASKEY

ASKEY, flowcharted in figure 4-11 and listed in appendix A, sheet 14, converts the four least significant bits (LSBs) in the accumulator from packed BCD to ASCII code with odd parity. The four most signficant bits (MSBs) in the accumulator are lost. The resultant ASCII character remains in the accumulator when control is returned to the calling routine. The Tektronix assembler does not permit the use of the word ASCII as a symbol. Therefore, this subroutine was called ASKEY.

4.2.10 PAUSE

This subroutine, shown in the flowchart of figure 4-12 and listed in appendix A, sheet 15, inserts a delay when it is called. The principal use of this subroutine is to create a pause between messages, but it could be used to create a pause anywhere, as long as the pause is equal to a whole number of seconds.

The number stored at location BFF0 in packed BCD form is used to determine how many times a one-second delay loop is executed. The operator enters the number BFF0 as the number of seconds of delay between messages.

The main delay occurs in lines 400 through 407 of the assembly listing. These lines are reproduced in table 4-1 along with the number of machine cycles required by each instruction. The XTHL instruction, which exchanges the contents of the HL register pair with the top of the stack, requires 18 clock cycles for execution. This is the longest time required by any 8085 instruction.

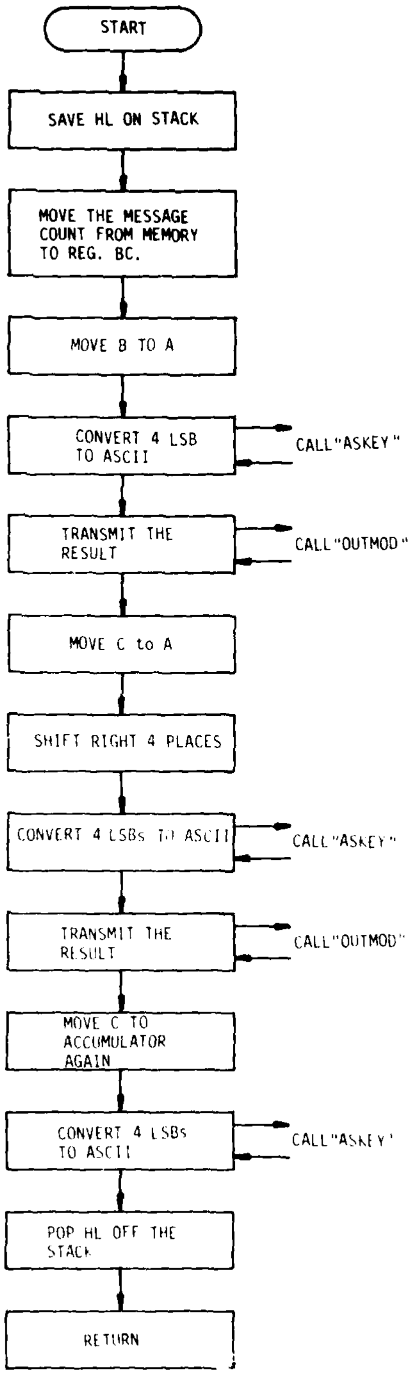


Figure 4-10. MSGNMBR Subroutine

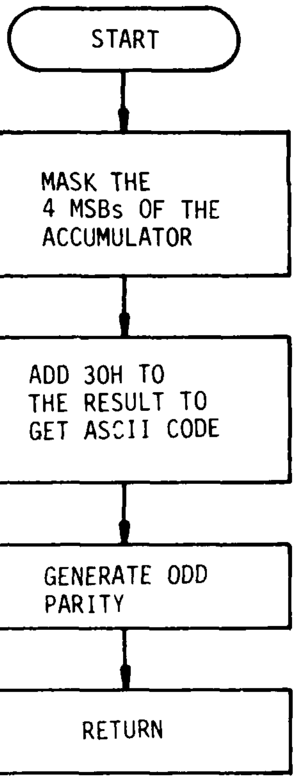

START
MASK THE 4 MSBs OF THE ACCUMULATOR
ADD 30H TO THE RESULT TO GET ASCII CODE
GENERATE ODD PARITY
RETURN

Figure 4-11. ASKEY Subroutine

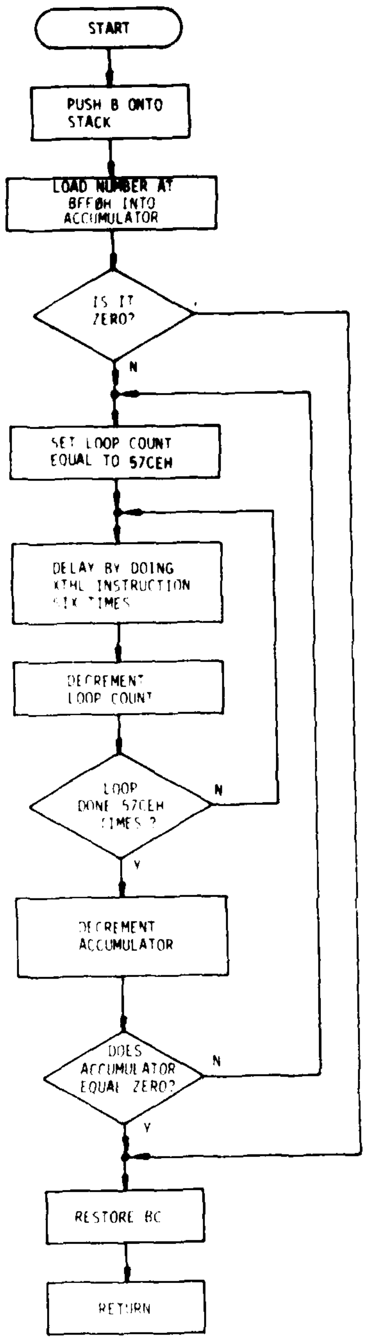

START
PUSH B ONTO STACK
LOAD NUMBER AT BFFØH INTO ACCUMULATOR
IS IT ZERO?
N
SET LOOP COUNT EQUAL TO 57CEH
DELAY BY DOING XTHL INSTRUCTION SIX TIMES
DECREMENT LOOP COUNT
LOOP DONE 57CEH TIMES ?
N
Y
DECREMENT ACCUMULATOR
DOES ACCUMULATOR EQUAL ZERO?
N
Y
RESTORE BC
RETURN

Figure 4-12. PAUSE Subroutine

Table 4-1

Delay Loop Timing

Instruction	Machine States
XTHL	18
XTHL	18
XTHL	18
XTHL	18
XTHL	18
XTHL	18
DCRC	5
JNZ	10
	123 machine states

If two XTHL instructions are executed in succession, the net result is that 36 clock periods have elapsed and nothing else. This is in contrast to an NOP instruction which requires only four clock periods to do nothing. Thus a pair of XTHL instructions makes an excellent delay.

Note from table 4-1 that the six XTHL instructions, a DCR instruction, and a JNZ instruction require a total of 123 machine states (clock periods). At 0.36168981 μs per machine state, this sequence of instructions will execute in 44.4878472 μs. Doing this loop 22,478 times would require nearly one second. 22,478 is expressed as 57CE in hexadecimal. Therefore, in line 399, the value 57CEH is loaded into register pair BC. This value is decremented once each time the delay loop is executed and when it reaches zero, one second has elapsed.

4.2.11 EPARITY

To transmit a character with even parity in the test message, the character to be sent with even parity must be preceded by a US (control /) character. When the XMTMSG subroutine encounters US, it calls EPARITY, which is listed in appendix A, sheet 16. The flowchart is shown in figure 4-13.

EPARITY increments the HL pair to address the next character of the message, fetches that character, generates even parity, and returns to the calling routine (XMTMSG), which will output the character thus generated.

4.2.12 PRTMSG

PRTMSG, shown in the flowchart of figure 4-14 and the listing in appendix A, sheet 19, is used to print the contents of a table in memory on the system console. Prior to calling PRTMSG, the calling routine must load the HL register pair with the address of the first byte of the table to be printed. Printing continues until a 00H is encountered. An even parity NUL (00H) is used to denote the end of all tables in memory. PRTMSG does not send the NUL character to the printer. The NUL is used to cause program execution to return to the calling routine.

4.2.13 TOGGLE2

TOGGLE, used to output random data between messages, is described in section 4.2.5 and should not be confused with TOGGLE2, which is described here.

TOGGLE2 (see flowchart, figure 4-15 and listing, appendix A, sheet 20) is called by XMTMSG whenever an FS character (control ",") is encountered in the test message. FS must be followed by two decimal digits. When TOGGLE2 is called, the two digits following the FS are read, converted from ASCII code to packed BCD, and set equal to N. The TOGGLE table (TA5) is then sent N times.

The first eight blocks of the flowchart deal with reading the two digits which represent the number N from th table containing the message, and converting N to packed BCD form. This is done in lines 491 through 505 of the program listing (appendix A, sheet 20). If N is non-zero, table TA5 is sent N times. The test for N = 0 is needed to keep the table from being sent 99 times if the operator enters 00.

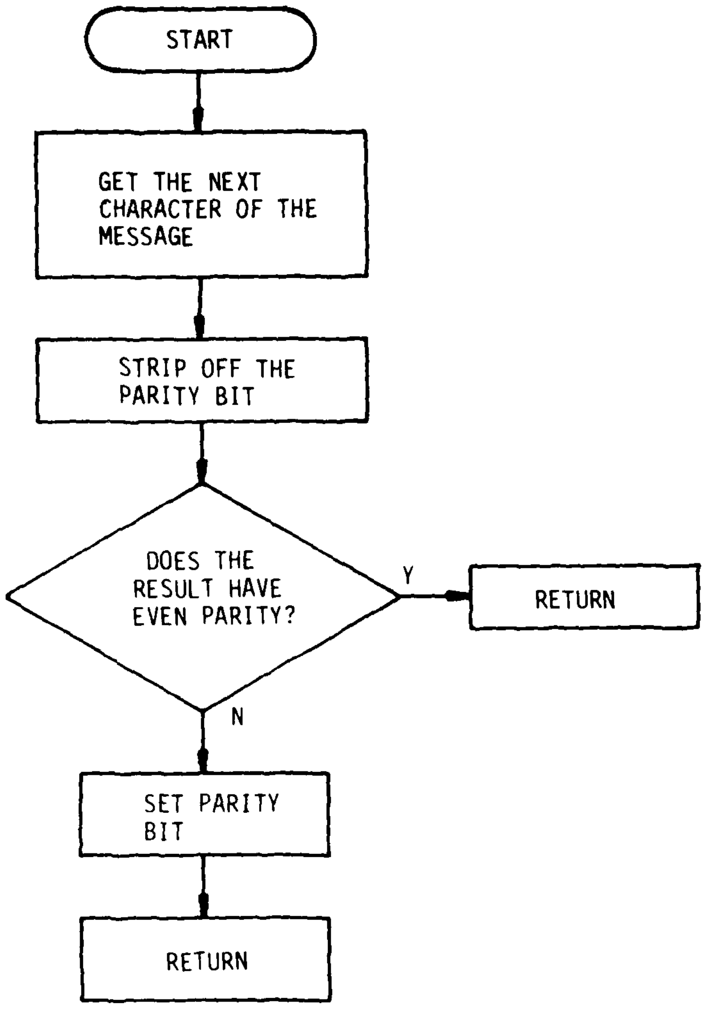

START
GET THE NEXT CHARACTER OF THE MESSAGE
STRIP OFF THE PARITY BIT
DOES THE RESULT HAVE EVEN PARITY?
Y
RETURN
N
SET PARITY BIT
RETURN

Figure 4-13. EPARITY Subroutine

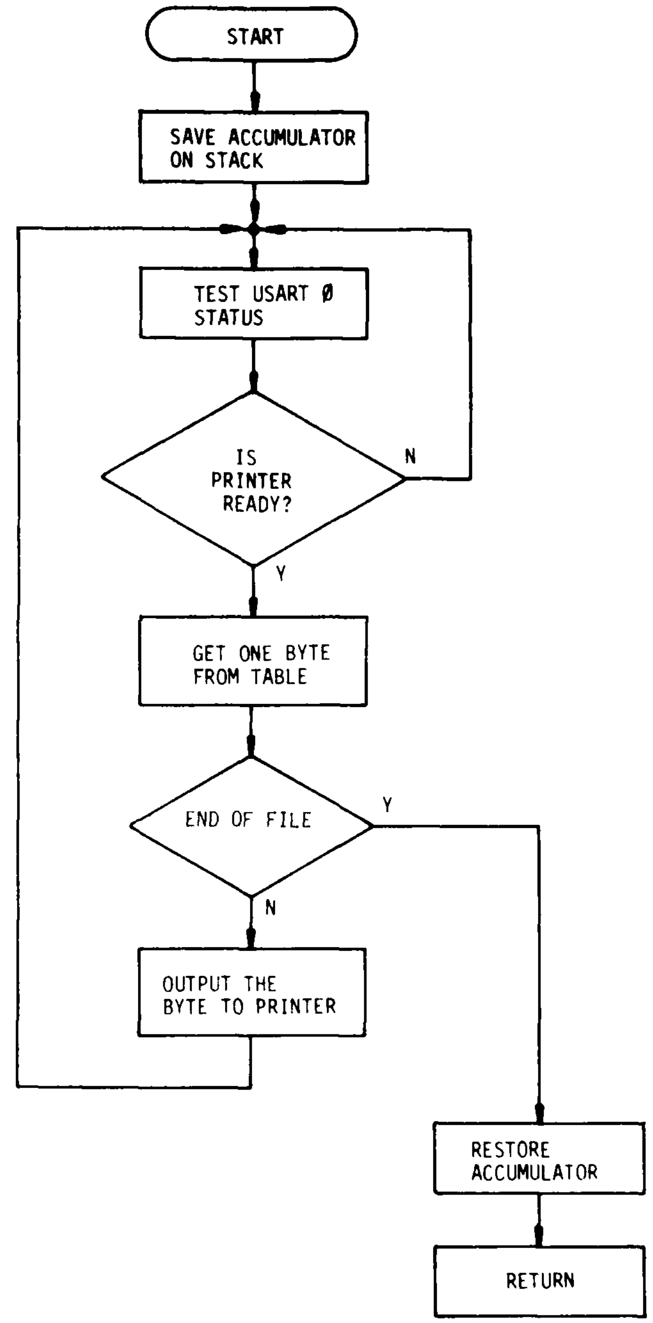


Figure 4-14. PRTMSG Subroutine

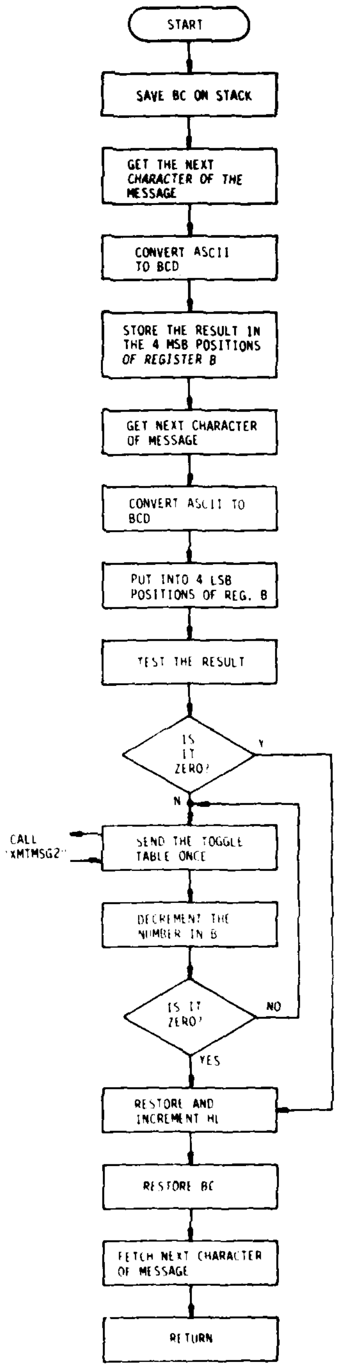

START
SAVE BC ON STACK
GET THE NEXT CHARACTER OF THE MESSAGE
CONVERT ASCII TO BCD
STORE THE RESULT IN THE 4 MSB POSITIONS OF REGISTER B
GET NEXT CHARACTER OF MESSAGE
CONVERT ASCII TO BCD
PUT INTO 4 LSB POSITIONS OF REG. B
TEST THE RESULT
IS IT ZERO?
Y
N
CALL "XMTMSG2"
SEND THE TOGGLE TABLE ONCE
DECREMENT THE NUMBER IN B
IS IT ZERO?
NO
YES
RESTORE AND INCREMENT HL
RESTORE BC
FETCH NEXT CHARACTER OF MESSAGE
RETURN

Figure 4-15. TOGGLE2 Subroutine

Decrementing of the count in register B is accomplished in lines 510 through 514 of the listing (appendix A, sheet 21). Since this is a BCD decrement, the register contents must be adjusted after the decrement to represent two valid BCD digits. This can be done with the decimal adjust accumulator (DAA) instruction. DAA makes use of the auxiliary carry (AC) bit, which indicates a carry out of the least significant BCD digit (four LSB positions) of the accumulator. The AC flag is properly set for the DAA instruction only following an ADD instruction. Therefore, decrementing of the count is accomplished by adding 99 to the count. 99 is the twos complement representation of -1. Therefore, adding 99 is the same as subtracting 1 as far as the contents of the register are concerned. The difference is in the state of the AC flag, which is needed for proper operation of the DAA instruction.

4.2.14 MSGIN

The flowchart in figure 4-16 is for the MSGIN subroutine. The listing appears in appendix A, sheet 22.

MSGIN first prints the prompt in table TA6. This subroutine then begins entering characters from the keyboard and storing them in memory, beginning at location 8000 (figure 4-1b). All characters except RS and BS are stored exactly as received by the computer from the system console. Not even the parity is changed. Therefore the system console must be set to generate odd parity.

If the operator makes an error in entering the message, a backspace (BS or control H) causes the buffer pointer (41 register pair) to be decremented. This allows the operator to type over an error in memory just as a typist would backspace to type over an error on an ordinary word processor. Typing RS (control ".") causes a branch back to the beginning of the MSGIN subroutine, in order to correct more serious errors. In this case, the prompt instructing the operator to enter the test message would be repeated and message input must begin again.

Message input is terminated by typing an ETX (control C) or an EOT (control D).

4.2.15 MSGCOUNT/DELAYIN

In the main program, this subroutine is labeled simply "MSGCOUNT" (line 581, appendix A). DELAYIN follows MSGCOUNT as part of the same subroutine, since no reason could be found to divide them into separate subroutines. MSGCOUNT inputs both the number of

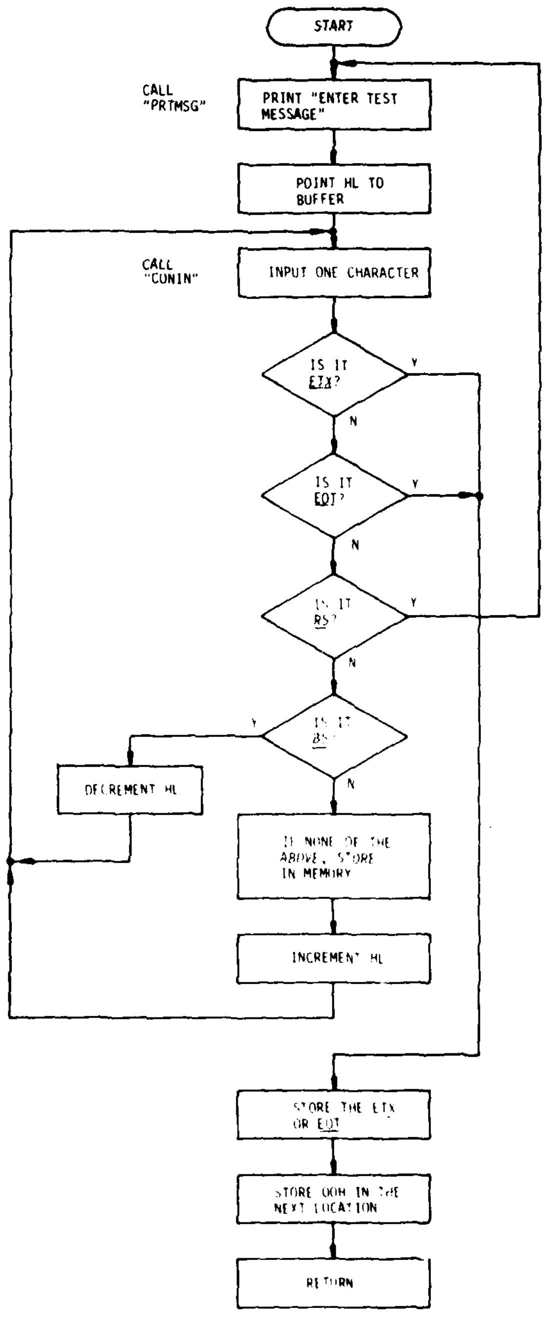


Figure 4-16. MSGIN Subroutine

messages in the test and the delay time between messages. The delay is entered in the second half of the subroutine and is given the symbolic address DELAYIN.

MSGCOUNT/DELAYIN is shown in the flowcharts of figures 4-17a and 4-17b. The listing appears on sheet 23 of appendix A.

MSGCOUNT first calls the PRTMSG subroutine to print table TA7 on the console. TA7 contains the prompt message which asks the operator to enter the number of times the message is to be sent.

Next, register pair DE is cleared. The number of times the message is to be transmitted will be stored in register pair DE in packed BCD form.

After clearing DE, the subroutine BCDIN is used to input one BCD digit. If the character typed is not a number between 0 and 9, BCDIN returns FFH in the accumulator. The FF signals the MSGCOUNT subroutine that an invalid character was typed and that the inputting of this test parameter should begin anew.

If the typed character is a carriage return, BCDIN returns it unchanged. This is the signal that the operator has completed entering the test parameter, and causes program execution to be transferred to the DELAYIN portion of the subroutine.

If the typed character is a digit between 0 and 9, the DE register pair is shifted four places to the left and the newly entered number is placed into the right four LSB positions. This process continues until the carriage return is typed. Thus, at the end of data entry, the DE pair is left containing the last four digits typed in packed BCD form. If fewer than four digits are typed prior to the carriage return, the most significant digits are zero, since the DE pair was cleared prior to the beginning of data entry.

DELAYIN first prompts the operator to enter the delay time between messages, then inputs BCD digits and stores the two most recently typed digits in register C in packed BCD form.

Except for using register C instead of the DE pair, and therefore keeping only the last two digits entered instead of the last four, DELAYIN works in a manner that is identical to MSGCOUNT.

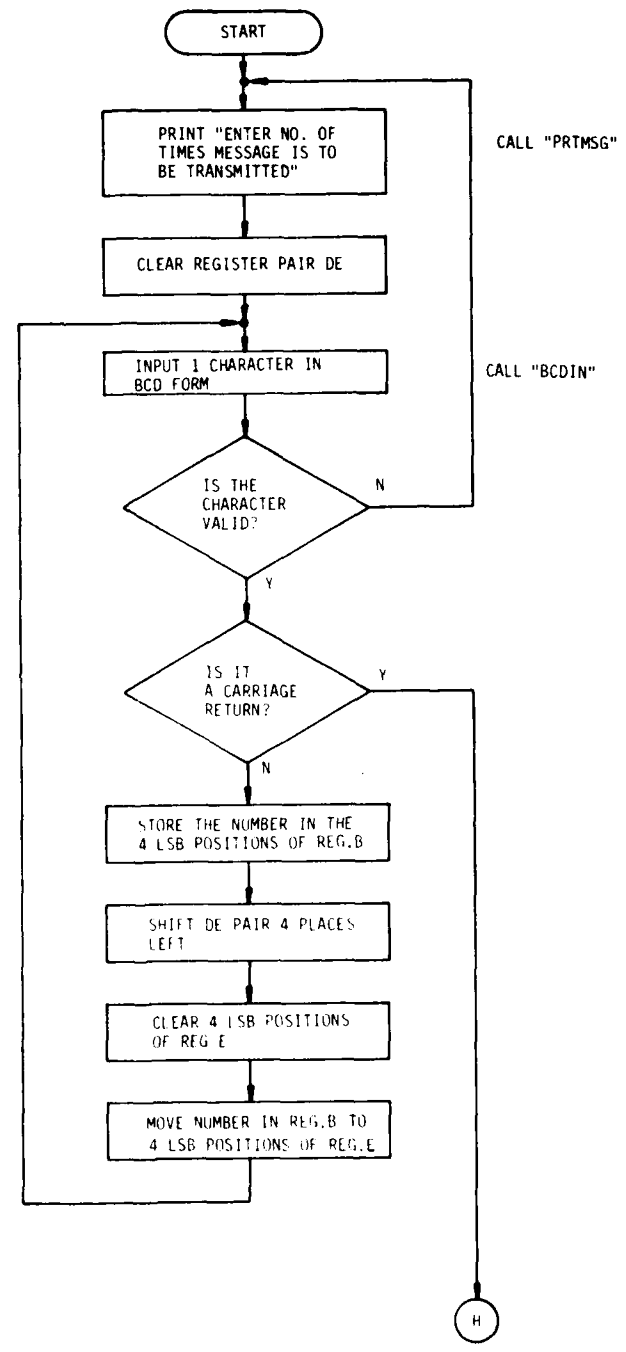

START
PRINT "ENTER NO. OF TIMES MESSAGE IS TO BE TRANSMITTED"
CALL "PRTMSG"
CLEAR REGISTER PAIR DE
INPUT 1 CHARACTER IN BCD FORM
CALL "BCDIN"
IS THE CHARACTER VALID?
N
Y
IS IT A CARRIAGE RETURN?
Y
N
STORE THE NUMBER IN THE 4 LSB POSITIONS OF REG.B
SHIFT DE PAIR 4 PLACES LEFT
CLEAR 4 LSB POSITIONS OF REG E
MOVE NUMBER IN REG.B TO 4 LSB POSITIONS OF REG.E
H

Figure 4-17a. MSGCOUNT Subroutine

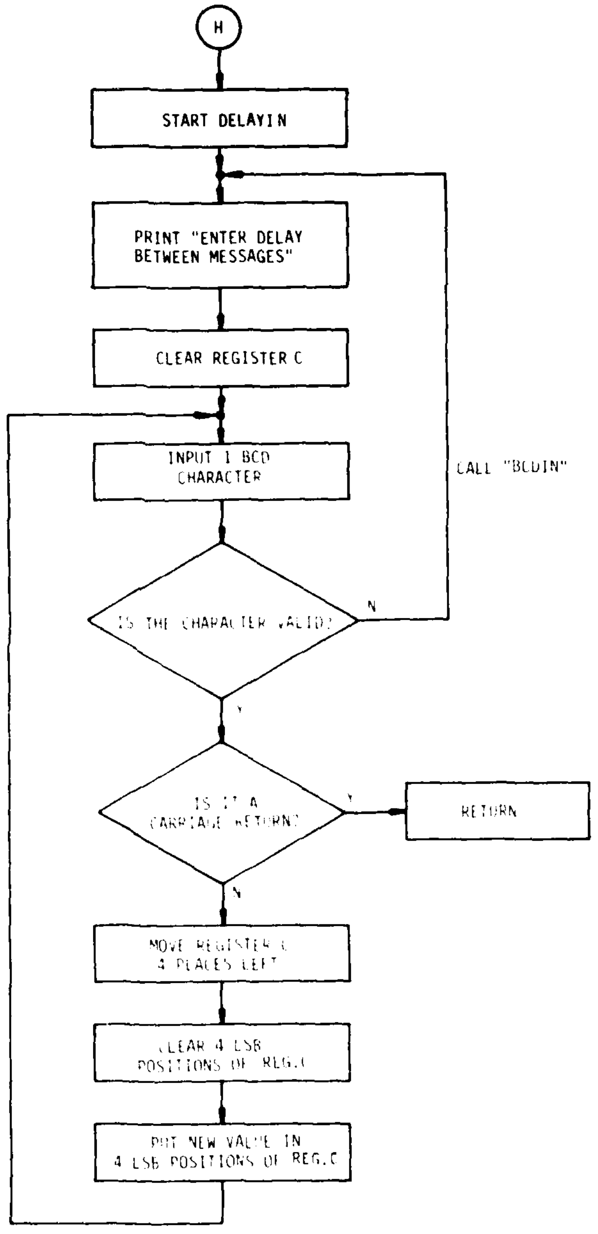

H
START DELAYIN
PRINT "ENTER DELAY BETWEEN MESSAGES"
CLEAR REGISTER C
INPUT 1 BCD CHARACTER
CALL "BCDIN"
IS THE CHARACTER VALID?
N
Y
IS IT A CARRIAGE RETURN?
Y
RETURN
N
MOVE REGISTER C 4 PLACES LEFT
CLEAR 4 LSB POSITIONS OF REG.C
PUT NEW VALUE IN 4 LSB POSITIONS OF REG.C

Figure 4-17b. DELAYIN Subroutine

4.2.16 BCDIN

BCDIN (see figure 4-18 and appendix A, sheet 25) inputs one character from the keyboard and tests it. If the operator typed a carriage return, it is left in the accumulator unchanged and returned to the calling routine. If the typed character is anything except a carriage return or a number between 0 and 9, FFH is returned to the calling routine in the accumulator.

If the typed character is a number between 0 and 9, it is converted from ASCII to BCD and returned to the calling routine in the four LSB positions of the accumulator. The four MSB positions of the accumulator are cleared.

Testing the typed character for a valid input consisting of a number between 0 and 9 is quite simple. In ASCII code the digits 0 through 9 are represented by 30H through 39H, respectively. Therefore, testing whether or not the typed character is between these limits will reveal whether it is a digit or not.

Once the typed character is determined to be a digit, stripping off the four most significant bits converts the digit from ASCII code to binary. This is returned to the calling routine in the accumulator.

4.2.17 CLEARCOUNT

CLEARCOUNT is used at the beginning of a test sequence to clear memory location BFF2 and BFF3. These two locations store the running count of the number of messages which have been transmitted. The count is incremented and tested at the end of each message transmission to determine if the desired number of messages has been sent. CLEARCOUNT is listed in appendix A, sheet 26, and the flowchart is shown in figure 4-19.

4.2.18 PORT2OUT/PORT3OUT

These subroutines output a table from memory via serial ports 2 and 3. They were included to facilitate future expansion. Serial I/O ports 2 and 3 are not currently used and neither are these two subroutines.

The flowcharts are shown in figures 4-20a and 4-20b, and the listings appear in appendix A, pages 28 and 29. These subroutines operate identically. Before either one is called, the HL register pair must be loaded with the address of the first byte in the table

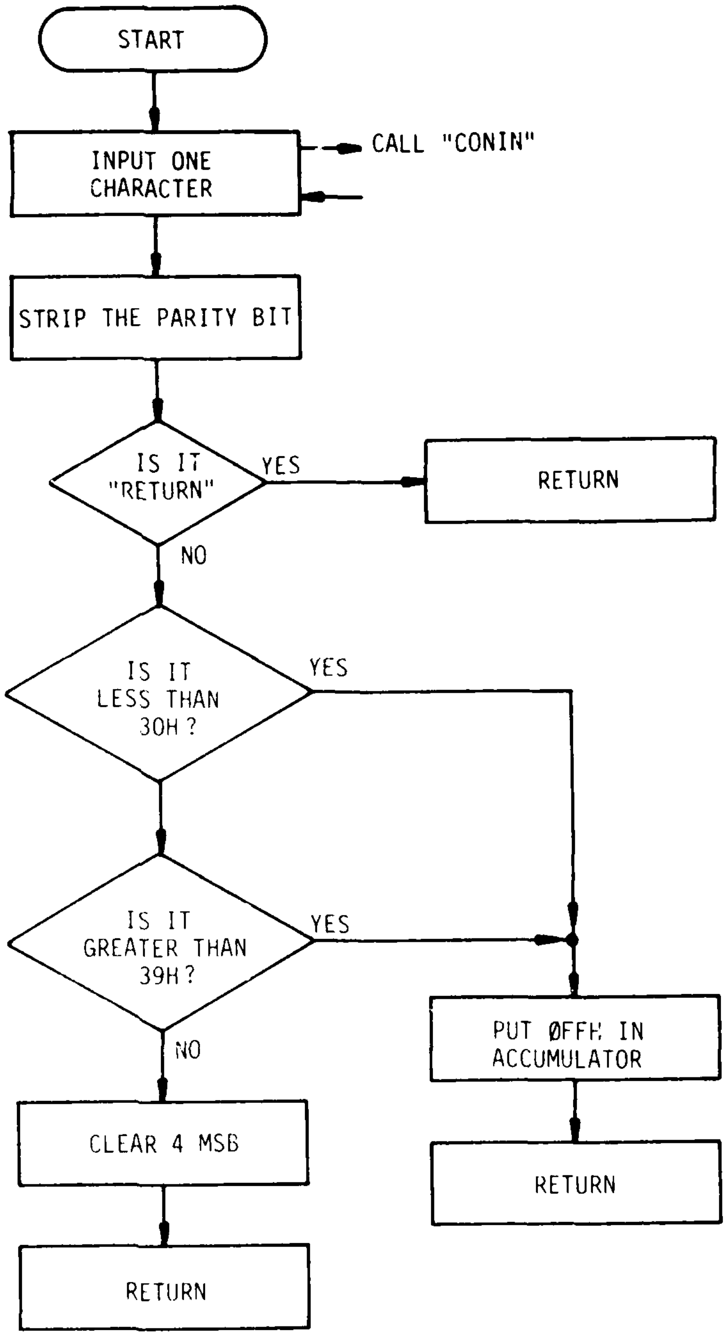


Figure 4-18. BCDIN Subroutine

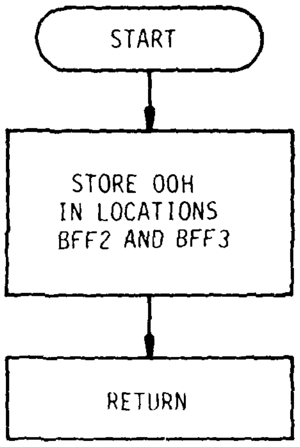


Figure 4-19. CLEARCOUNT Subroutine

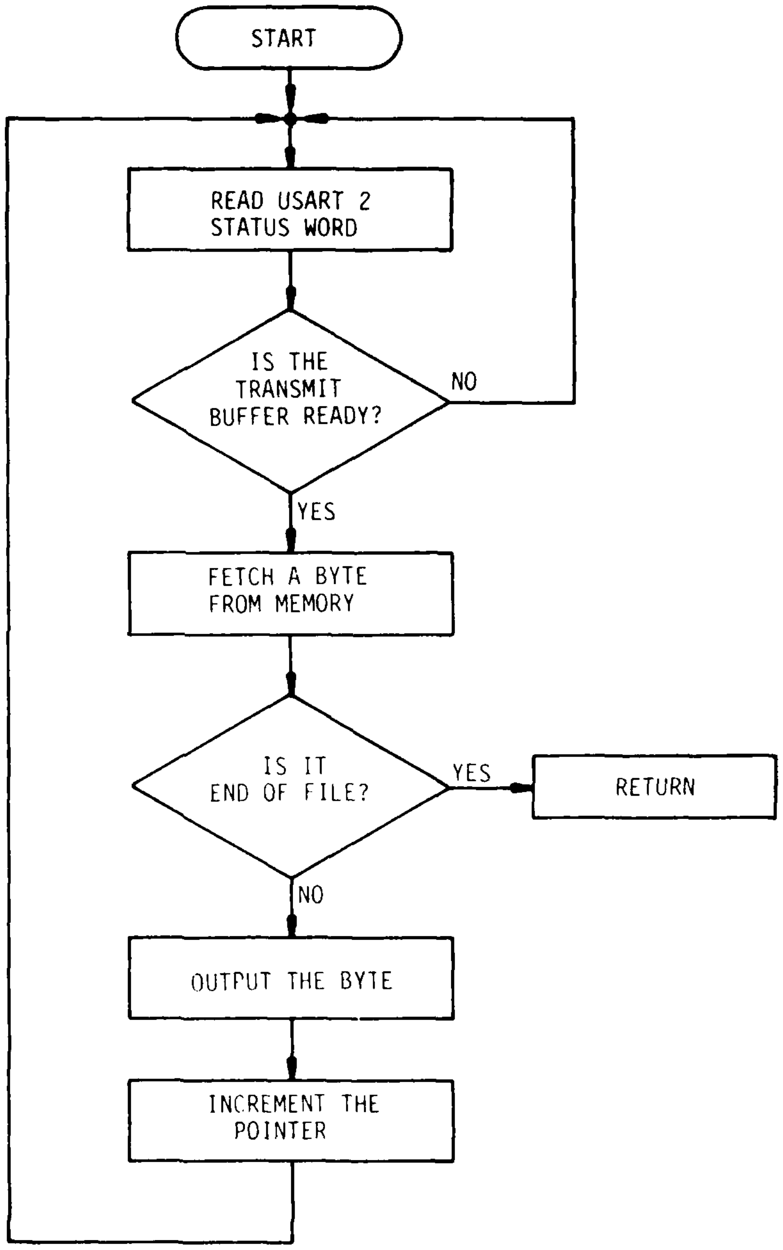


Figure 4-20a. PORT2OUT Subroutine

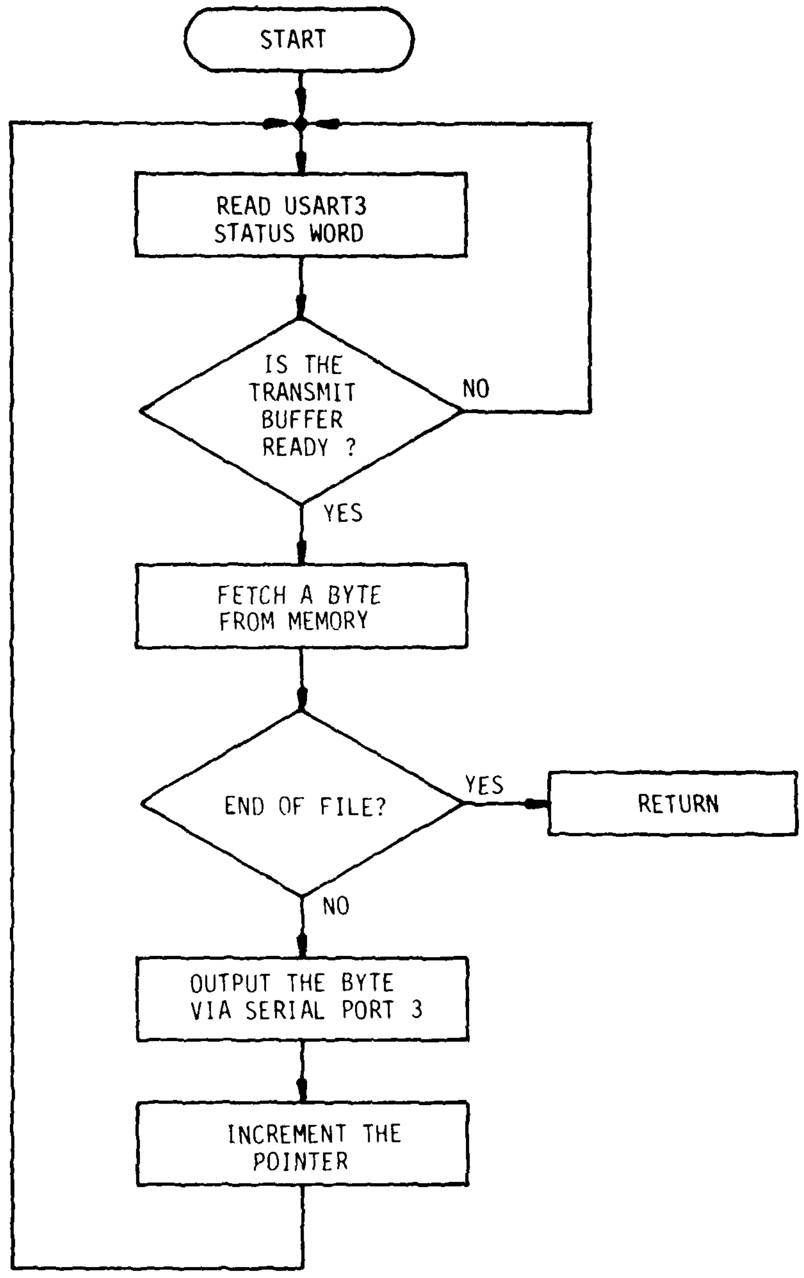


Figure 4-20b. PORT3OUT Subroutine

to be outputted. All characters in the table will be outputted in turn until a 00H is encountered, marking the end of the table.

4.2.19 CHARIN2/CHARIN3

CHARIN2 and CHARIN3 are not used currently, but are provided as utility routines to facilitate expansion. Their purposes are to input one character via serial I/O port 2 or 3, and return it to the calling routine in the accumulator. The flowcharts are shown in figures 4-21a and 4-21b. The program listings appear on sheets 30 and 31 of appendix A.

These subroutines, once called, test the appropriate USART status word until the status word indicates that a character has been received via the associated serial I/O port. When this occurs, the word is read from the input buffer and is returned, in the accumulator, to the calling routine.

4.2.20 RCVMSG

RCVMSG accepts a received message from an AFSATCOM modem, prints each character as it is received on the system console, and stores the message in memory.

RCVMSG was initially intended as a utility subroutine, as the message controller has no current need to process received messages. Since it was considered desirable to test both the receive hardware and the RCVMSG subroutine, the ASRSIM program was written. However, RCVMSG is still basically an unused subroutine included for future expansion if needed. The flowchart is shown in figure 4-22 and the listing appears in appendix A, sheet 32.

Before calling RCVMSG, it is necessary to load the HL register pair with the address in memory where the first character of the received message is to be stored.

For proper operation, it is also necessary to determine that a message is being received by the AFSATCOM modem prior to calling RCVMSG. This is done by testing I/O port 0EBH. If bit 1 is a logic "1", the bit clock is running, and RCVMSG should be called.

If RCVMSG is called when the bit clock is not running, control of program execution returns to the calling routine almost immediately. However, an end-of-file marker will be stored in the memory and a carriage return/line feed will be sent to the printer.

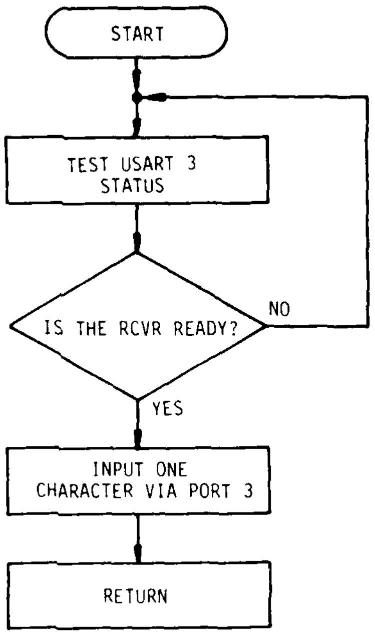


Figure 4-21a. CHARIN2

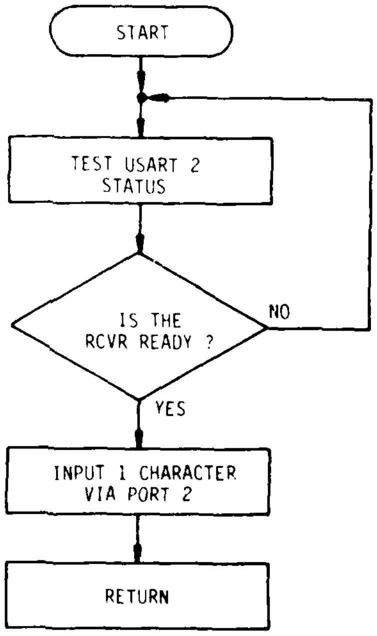


Figure 4-21b. CHARIN3

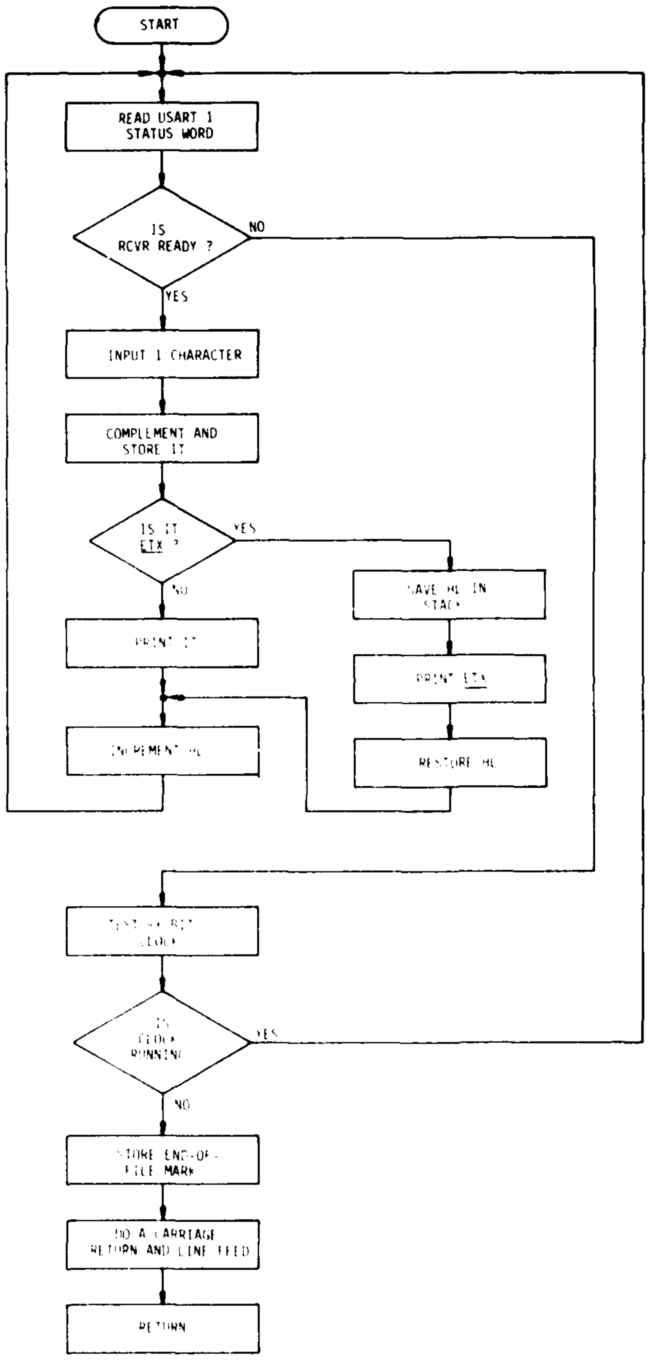


Figure 4-22. RCVMSG Subroutine

Since an ETX is not mandatory at the end of an AFSATCOM message, RCVMSG uses the bit clock to determine whether or not the entire message has been received. When the bit clock stops, message transmission is assumed to be complete.

Operation of the subroutine is quite straightforward. As long as the receive bit clock from the modem is running, the program waits until a complete character has been received from the modem. The received character is then read, complemented, and stored in memory. If the received character is not an ETX,it is printed before the next character is fetched.

NOTE: ETX is not a printing character on the TI-765 printer.

If the received character is an ETX, the letters E, T, X, and a space are sent to the printer in lieu of the ETX. The printer can operate at about three times the AFSATCOM rate. Four characters are printed for each ETX character received. Therefore, the printer runs slightly behind when ETX characters are being received. This slack is taken up by the one character (8 bit) buffer in the USART.

If a long string of ETX characters is imbedded in the message, occasional characters will be lost after about three or four ETX characters. Since AFSATCOM equipment does not readily allow more than two odd parity ETX characters to be sent, and only sends them at the end of a message, this is not a problem.

The even parity ETX characters appended by the modem are not printed. When the complete message has been received, i.e., when the bit clock stops, a 00H is stored in memory to denote the end of the message and a carriage return/line feed is sent to the printer.

4.3 ASRSIM PROGRAM

ASRSIM might be called an alternative executive program. The flowchart is shown in figure 4-23 and the listing is on sheet 34 of appendix A.

ASRSIM does a partial emulation of the AFSATCOM ASR. Messages can be entered from the keyboard and transmitted, and messages can be received and printed. The ESC key acts as the AUTO XMT key does on the AFSATCOM ASR. ASRSIM allows no manual transmit, poll transmit, verify, or selective addressing. This program was included primarily to test the receive hardware and to test and debug the RCVMSG utility subroutine.

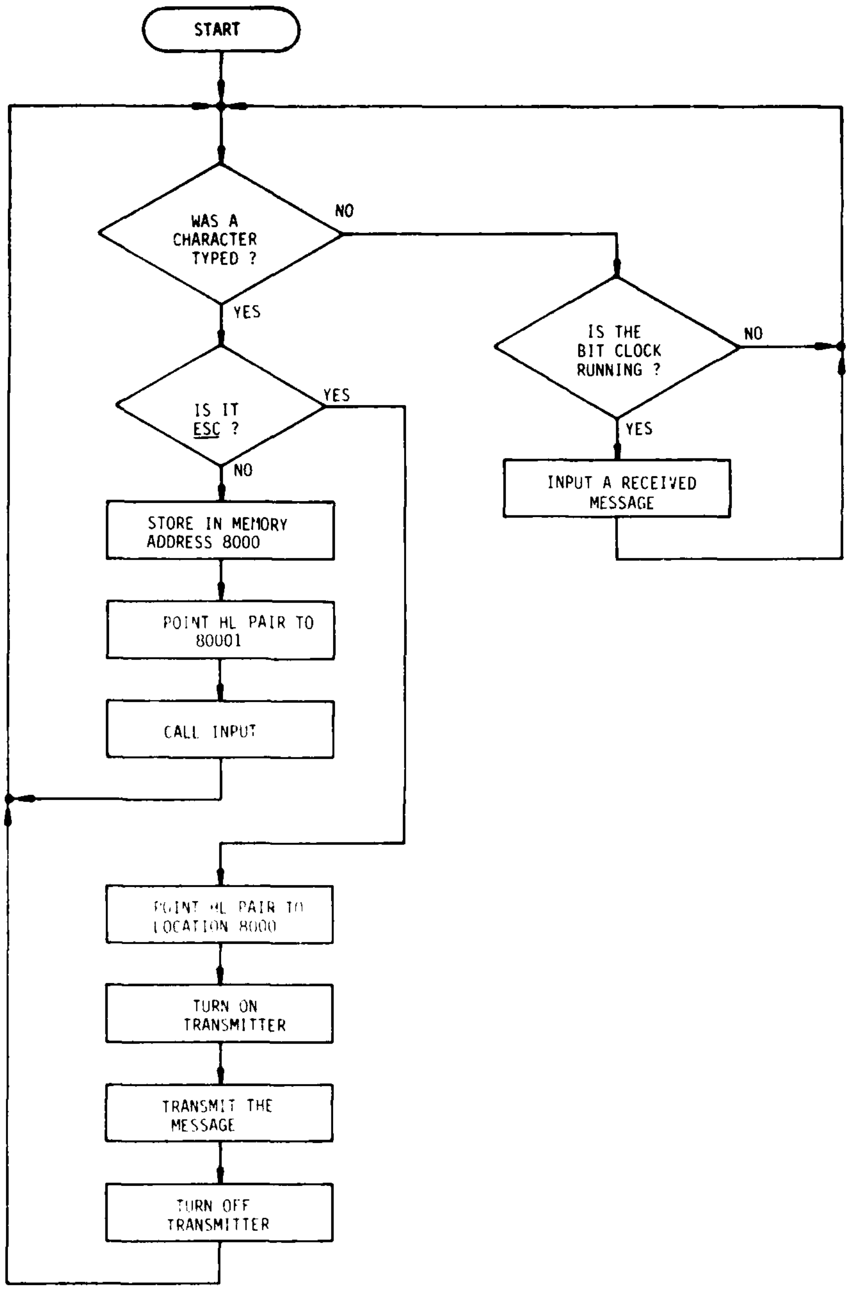


Figure 4-?3. ASRIM Program

ASRSIM is entered by typing "A" in repsonse to the prompt "REPETITIVE MESSAGE TEST, ASR EMULATION, OR SPECIAL TEST? TYPE R/A/S." Once ASRSIM is running, the only way to exit the program is to depress the front panel reset button.

Refer to the flowchart. Once ASRSIM is running, it enters a wait state where it alternately tests the system console and the receive bit clock.

When the bit clock starts, the RCVMSG subroutine is called. RCVMSG prints the message as it is being received: when the message ends and control returns to ASRSIM, nothing is done with the received message stored in memory. Once the bit clock starts, no message can be entered via the keyboard until the bit clock has stopped and ASRSIM has returned to the wait state.

If a character is typed while ASRSIM is in the wait state, the character will be tested. If it is an escape ($\overline{\text{ESC}}$), the message previously entered via the keyboard will be transmitted. If the character is anything other than ESC, the program assumes it is the first character of a new message and stores it at location 8000. A subroutine called INPUT then inputs the rest of the message. Input is really the MSGIN subroutine, but by entering it at the point labeled INPUT, (line 540, sheet 22, appendix A) the part of the subroutine which prints the operator prompt and loads the HL pair is bypassed.

When the MSGIN subroutine returns control to ASRSIM, ASRSIM enters the wait state. The message is not transmitted until the $\overline{\text{ESC}}$ key is depressed. Thus the $\overline{\text{ESC}}$ key emulates the action of the $\overline{\text{AUTO XMT}}$ key on the AFSATCOM ASR.

Any received message which begins while a message is being entered from the keyboard or while a message is being transmitted will be lost.

4.4 TABLES

The final lines of the program listing on sheet 35 of appendix A reserve blocks of memory for the tables. Tables TA1, TA2, TA6, TA7, and TA8 are all operator prompts.

Table TA3 contains the WU $\overline{\text{SYN SYN}}$ preamble which is transmitted before the message in the regenerative channel simulation mode.

Table TA4 contains the four even parity ETX characters which must be transmitted immediately following the message in the regenerative channel simulation mode.

Table TA5 contains the 10-character table used by both the TOGGLE and TOGGLE2 subroutines.

Table TA9 contains the carriage return and line feed which is sent to the printer at various times to insure that everything starts printing at the left edge of the paper.

Table TA10 contains the characters E, T, X and space which are sent to the printer by the RCVMSG subroutine in lieu of the ETX character.

APPENDIX A

PROGRAM LISTING

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                 Sheet 1
INITIALIZE BAUD TIMER

                                               ;
      0000    >         ORG         000H       ;START AT LOCATION 000H.
                                               ;COMMAND PIO TO REMOVE RESET FROM USARTS
                                               ;BY:
 0000 3E41              MVI         A,41H      ;OUTPUTTING MODE WORD 41H TO
 0002 D3E8              OUT         0E8H       ;PORT E8H AND
 0004 3EC0              MVI         A,0C0H     ;THEN OUTPUTTING COMMAND WORD
 0006 D3E9              OUT         0E9H       ;0C0H TO PORT 0E9H.
                                               ;
                                               ;SET SERIAL I/O BAUD RATE TIMERS FOR
                                               ;PORTS 0,2,AND 3 TO OPERATE AT
                                               ;300 BAUD.  (TIMERS DIVIDE BY 0040H)
                                               ;
 0008 3E36              MVI         A,0036H    ;THE MODE WORD FOR TIMERS FOR SERIAL PORTS
 000A D3DB              OUT         0DBH       ;0 AND 3 IS 36H. OUTPUT 36H TO 0DBH
 000C D3DF              OUT         0DFH       ;AND 0DFH.
                                               ;
 000E 3EB6              MVI         A, 0B6H    ;THE MODE WORD FOR TIMER FOR SERIAL PORT 2
 0010 D3DB              OUT         0DBH       ;IS 0B6H. OUTPUT 0B6H TO 0DBH.
                                               ;
 0012 3E40              MVI         A,40H      ;LOAD COMMAND WORD 040H (LSB) INTO A AND
 0014 D3D8              OUT         0D8H       ;OUTPUT TO 0D8H,
 0016 D3DA              OUT         0DAH       ;TO 0DAH,
 0018 D3DC              OUT         0DCH       ;AND TO 0DCH.
                                               ;
 001A 3E00              MVI         A,00H      ;LOAD COMMAND WORD MSB (00H) INTO A
 001C D3D8              OUT         0D8H       ;AND OUTPUT TO 0D8H,
 001E D3DA              OUT         0DAH       ;TO 0DAH,
 0020 D3DC              OUT         0DCH       ;AND TO 0DCH.
```

Tektronix 8080/8085 ASM V3.3 MESSAGE CONTROLLER VERSION 4   Sheet 2
INITIALIZE USARTS

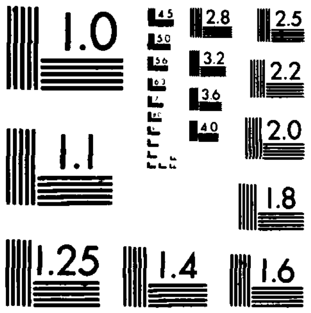
```
                                          ;SET USART 0 FOR X64 BAUD RATE, ASYN-
                                          ;CHRONOUS MODE, TX ENABLED, RCV ENABLED,
                                          ;8 BIT WORD, PARITY DISABLED.
                                          ;
                                          ;SET USARTS 2 AND 3 FOR X64 BAUD RATE,
                                          ;ASYNCHRONOUS MODE, TX ENABLED, RCV ENABLED,
                                          ;7 BIT WORD, EVEN PARITY.
                                          ;
0022 3E4F          MVI      A,04FH        ;USART 0 MODE WORD IS 04FH.
0024 D3D1          OUT      OD1H          ;OUTOUT MODE WORD TO OD1H.
0026 3E7B          MVI      A,07BH        ;USARTS 2 AND 3 MODE WORD IS 07BH.
0028 D3D5          OUT      OD5H          ;OUTPUT MODE WORD TO OD5H AND OD7H.
002A D3D7          OUT      OD7H          ;
002C 3E37          MVI      A,037H        ;COMMAND WORD FOR USARTS 0,2,AND 3 IS 37H.
002E D3D1          OUT      OD1H          ;OUTPUT COMMAND WORD TO OD1H,
0030 D3D5          OUT      OD5H          ;TO OD5H,
0032 D3D7          OUT      OD7H          ;AND TO OD7H.
                                          ;SET MODEM USART (#1) TO SYNCHRONOUS MODE,
                                          ;TX ENABLED, RCV ENABLED, RTS = 0,
                                          ;DTR = 0, ENTER HUNT, SYNCH WORD = $.
0034 3ECC          MVI      A, OCCH       ;MODE WORD = OCCH.
0036 D3D3          OUT      OD3H          ;OUTPUT MODE WORD TO OD3H.
0038 00            NOP                    ;WAIT
0039 00            NOP                    ;WAIT
003A 3E5B          MVI      A,05BH        ;SYNCH WORD IS 05BH.
003C D3D3          OUT      OD3H          ;OUTPUT SYNCH WORD TO OD3H.
003E 00            NOP                    ;WAIT
003F 00            NOP                    ;WAIT
0040 3E94          MVI      A,094H        ;COMMAND WORD IS 094H.
0042 D3D3          OUT      OD3H          ;OUTPUT COMMAND WORD TO OD3H.
0044 00            NOP                    ;
0045 00            NOP                    ;LEAVE SPACE FOR A JUMP.
0046 00            NOP                    ;
0047 31EFBF        LXI      SP,OBFEFH     ;INITIALIZE STACK POINTER TO OBFEFH.
```

AD-A126 262 TEST MESSAGE GENERATOR AND CONTROLLER FOR AFSATCOM TESTING(U) MITRE CORP BEDFORD MA D O ALWINE MAR 83 MTR-8840 F19628-82-C-0001

2/2

UNCLASSIFIED F/G 17/2 NL

END
DATE
FILMED
4 83
DTIC


1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

```
004A C35000  >             JMP      BEGIN      ;INITIALIZATION IS COMPLETE,BEGIN MAIN PROGRAM.
004D CD5D01  >  RESTART    CALL     XMTOFF     ;TURN OFF THE TRANSMITTER.
0050 21B603  >  BEGIN      LXI      H,TA1      ;POINT H TO PROMPT AND
0053 CD2302  >             CALL     PRTMSG     ;PRINT: "REPETITIVE MESSAGE TEST, ASR EMULATION,
                                               ;         OR SPECIAL TEST? TYPE R/A/S."
0056 CD8201  >             CALL     CONIN      ;CONIN INPUTS ONE CHARACTER FROM THE ASR.
0059 4F                    MOV      C,A        ;STORE A TEMPORARILY IN C.
005A 217005  >             LXI      H,TA9      ;
005D CD2302  >             CALL     PRTMSG     ;PRINT CR LF.
0060 79                    MOV      A,C        ;RESTORE A
0061 FED3                  CPI      0D3H       ;IS THE CHARACTER AN ASCII "S"?
0063 CA0008                JZ       0800H      ;IF S, JUMP TO LOC 0800H.
0066 FEC1                  CPI      0C1H       ;IS IT AN ASCII "A"?
0068 CA7603  >             JZ       ASRSIM     ;IF IT IS "A", DO AN AFSATCOM ASR EMULATION.
006B FE52                  CPI      052H       ;IS THE CHARACTER AN ASCII "R"?
006D C25000  >             JNZ      BEGIN      ;IF NONE OF THE ABOVE, TRY AGAIN.
                                               ;IF CHARACTER WAS AN "R", INPUT A TEST MESSAGE.
0070 CD6702  >  INPUTMSG   CALL     MSGIN      ;CALL THE TEST MESSAGE INPUT ROUTINE.
0073 CD9502  >             CALL     MSGCOUNT   ;CALL THE ROUTINE WHICH INPUTS THE
                                               ;NUMBER OF MESSAGES IN TEST
                                               ;AND THE DELAY BETWEEN MESSAGES.
0076 CD0203  >  CLEAR      CALL     CLEARCOUNT ;CLEAR THE MESSAGE COUNT.
0079 210104  >  READY      LXI      H,TA2      ;POINT HL TO THE PROMPT TABLE AND
007C CD2302  >             CALL     PRTMSG     ;PRINT: "TYPE R TO BEGIN REGEN TEST
                                               ;         TYPE N TO BEGIN NON REGEN TEST
                                               ;         TYPE A TO ABORT."
007F CD8201  >             CALL     CONIN      ;INPUT 1 CHARACTER FROM CONSOLE.
0082 FE52                  CPI      052H       ;IS THE CHARACTER AN R?
0084 CA1502  >             JZ       REGEN      ;IF SO, SET THE REGEN FLAG. (DTR BIT=1)
0087 FECE                  CPI      0CEH       ;IS THE CHARACTER AN N?
0089 CA1C02  >             JZ       NONREGEN   ;IF SO SET THE NONREG FLAG. (DTR=0)
008C FEC1                  CPI      0C1H       ;IS THE CHARACTER AN A?
008E CA4D00  >             JZ       RESTART    ;IF SO, ABORT THE TEST.
0091 C27900  >             JNZ      READY      ;IF NEITHER R, N, OR A, TRY AGAIN.
```

Tektronix 8080/8085 ASM V3.3 MESSAGE CONTROLLER VERSION 4 Sheet 4
MAIN PROGRAM

```
0094 DBD1          TESTCON   IN     OD1H       ;TEST USART1 STATUS WORD.
0096 E602                    ANI    02H        ;WAS A CHARACTER INPUT FROM CONSOLE?
0098 CAC800   >              JZ     SENDMSG    ;IF NOT SEND A TEST MESSAGE.
009B DBD0                    IN     ODOH       ;IF A CHARACTER WAS INPUT, READ IT.
009D D3D0          ECHO      OUT    ODOH       ;AND ECHO IT TO THE CONSOLE.
009F FE52                    CPI    52H        ;IS IT AN R?
00A1 CA7600   >              JZ     CLEAR      ;IF SO, BEGIN THE TEST AGAIN.
00A4 FEC1                    CPI    OC1H       ;IS IT AN A?
00A6 CA4D00   >              JZ     RESTART    ;IF SO, ABORT THE TEST.
00A9 FE20                    CPI    020H       ;IS IT A SPACE?
00AB C2C800   >              JNZ    SENDMSG    ;IF NOT A, R, OR SPACE, IGNORE IT
                                               ;AND SEND NEXT MESSAGE.
00AE DBD3          WAIT      IN     OD3H       ;READ MODEM USART STATUS WORD.
00B0 E680                    ANI    80H        ;IS THE TEST REGEN?
00B2 CABA00   >              JZ     WHATNEXT   ;IF NOT REGEN, HALT TEST UNTIL ANOTHER
                                               ;CHARACTER IS INPUTTED.
00B5 3E16                    MVI    A,16H      ;IF REGEN TEST, OUTPUT A SYN CHARACTER
00B7 CDB501   >              CALL   OUTMOD     ;TO AFSATCOM MODEM.
00BA DBD1          WHATNEXT  IN     OD1H       ;WAS A CHARACTER INPUT BY OPERATOR?
00BC E602                    ANI    02H        ;
00BE CAAE00   >              JZ     WAIT       ;IF NOT, OUTPUT ANOTHER SYN CHARACTER.
00C1 DBD0                    IN     ODOH       ;IF CHARACTER WAS TYPED, READ IT.
00C3 FE20                    CPI    20H        ;IS IT A SPACE?
00C5 C29D00   >              JNZ    ECHO       ;IF NOT A SPACE, ECHO IT AND RETEST.
00C8 CD4A01   >    SENDMSG   CALL   XMTON      ;TURN ON TRANSMITTER.
00CB DBD3                    IN     OD3H       ;READ MODEM STATUS WORD.
00CD E680                    ANI    080H       ;IS THIS A REGEN TEST?
00CF CADB00   >              JZ     SEND2      ;IF NOT REGEN, SKIP TOGGLE.
00D2 CD8E01   >              CALL   TOGGLE     ;IF REGEN TEST, SEND TOGGLE SEQUENCE.
00D5 215B04   >              LXI    H,TA3      ;POINT H TO THE PREAMBLE TABLE AND
00D8 CDAA01   >              CALL   XMTMSG2    ;SEND THE PREAMBLE.
00DB 210080        SEND2     LXI    H,8000H    ;POINT H TO START ADDRESS OF TEST MESSAGE.
```

```
00DE CD1D01  >              CALL      XMTMSG       ;AND SEND THE MESSAGE.
00E1 DBD3                   IN        0D3H         ;READ MODEM STATUS WORD.
00E3 E680                   ANI       80H          ;IS THIS A REGEN TEST?
00E5 C21101  >              JNZ       POSTAMBLE    ;IF SO, SEND POSTAMBLE AND TOGGLE.
00E8 CD5D01  >              CALL      XMTOFF       ;IF NON-REGEN, TURN OFF TRANSMITTER,
                                                   ;MODEM WILL SEND POSTAMBLE AUTOMATICALLY.
00EB CDE901  >              CALL      PAUSE        ;PAUSE BEFORE SENDING THE NEXT MESSAGE.
00EE 21F3BF     INCREMENT   LXI       H,0BFF3H     ;THEN INCREMENT THE MESSAGE COUNT
00F1 7E                     MOV       A,M          ;STORED AT LOCATIONS BFF2H AND BFF3H AS 4 DIGITS
00F2 C601                   ADI       01H          ;OF PACKED BCD.  LEAST SIGNIFIGANT BITS
00F4 27                     DAA                    ;IN BFF3.
00F5 77                     MOV       M,A          ;PUT THE RESULT BACK IN MEMORY.
00F6 D20001  >              JNC       TESTCOUNT    ;
00F9 2B                     DCX       H            ;
00FA 7E                     MOV       A,M          ;
00FB C601                   ADI       01H          ;
00FD 27                     DAA                    ;
00FE 77                     MOV       M,A          ;
00FF 23                     INX       H            ;
0100 7E         TESTCOUNT   MOV       A,M          ;NOW TEST THE MESSAGE COUNT TO SEE IF
0101 BB                     CMP       E            ;ANOTHER MESSAGE SHOULD BE SENT.
0102 C29400  >              JNZ       TESTCON      ;IF COUNT HAS NOT REACHED THE VALUE STORED
0105 2B                     DCX       H            ;IN REGISTERS D AND E, GO TO TESTCON TO
0106 7E                     MOV       A,M          ;TEST WHETHER OR NOT A CHARACTER HAS BEEN
0107 BA                     CMP       D            ;TYPED ON THE CONSOLE.
0108 C29400  >              JNZ       TESTCON      ;
010B CD5D01  >              CALL      XMTOFF       ;TURN OFF TRANSMITTER WHEN TEST IS COMPLETE.
010E C37600  >              JMP       CLEAR        ;
0111 216004  >  POSTAMBLE   LXI       H,TA4        ;POINT H TO POSTAMBLE TABLE.
0114 CDAA01  >              CALL      XMTMSG2      ;SEND THE POSTAMBLE.
0117 CD8E01  >              CALL      TOGGLE       ;SEND RANDOM DATA FOR N SECONDS.
011A C3EE00  >              JMP       INCREMENT    ;INCREMENT AND TEST THE MESSAGE COUNT.
```

```
                                                    ;THIS SUBROUTINE OUTPUTS A MESSAGE FROM
                                                    ;MEMORY TO THE AFSATCOM MODEM.  PRIOR TO
                                                    ;CALLING THIS SUBROUTINE THE TRANSMITTER
                                                    ;MUST BE TURNED ON WITH THE XMTON SUB-
                                                    ;ROUTINE AND THE HL PAIR MUST BE POINTED
                                                    ;TO THE LOCATION OF THE FIRST CHARACTER
                                                    ;OF THE MESSAGE.  CONTROL RETURNS TO
                                                    ;THE CALLING ROUTINE WHEN AN ETX OR EOT
                                                    ;CHARACTER IS ENCOUNTERED.
011D 7E          XMTMSG    MOV        A,M           ;FETCH THE FIRST CHARACTER.
011E FE1A                  CPI        01AH          ;IS IT SUB?
0120 CCC201  >             CZ         MSGNMBR       ;IF SUB, TRANSMIT THE MESSAGE NUMBER.
0123 FE1F                  CPI        01FH          ;IS IT US?
0125 CC0D02  >             CZ         EPARITY       ;IF US CALL EPARITY SUBROUTINE.
0128 FE1C                  CPI        01CH          ;IS IT FS?
012A CC3902  >             CZ         TOGGLE2       ;IF FS, SEND RANDOM BITS FOR N SECONDS.
012D FE04                  CPI        04H           ;IS IT EOT?
012F C8                    RZ                       ;IF EOT, END THE MESSAGE.
0130 FE83                  CPI        83H           ;IS IT ETX?
0132 CA4201  >             JZ         ETX           ;IF ETX, SEND THE ETX TWICE AND END.
0135 00                    NOP                      ;
0136 00                    NOP                      ;LEAVE SPACE FOR ANOTHER SPECIAL FUNCTION.
0137 00                    NOP                      ;
0138 00                    NOP                      ;
0139 00                    NOP                      ;
013A 00                    NOP                      ;
013B CDB501  >             CALL       OUTMOD        ;IF THE CHARACTER IS NONE OF THE ABOVE,
                                                    ;TRANSMIT IT, THEN
013E 23                    INX        H             ;POINT HL TO THE NEXT CHARACTER.
013F C31D01  >             JMP        XMTMSG        ;GO FETCH THE NEXT CHARACTER.
0142 CDB501  >   ETX       CALL       OUTMOD        ;OUTPUT THE ETX.
0145 2F                    CMA                      ;COMPLEMENT THE ACCUMULATOR (OUTMOD
                                                    ;COMPLEMENTED IT ONCE) AND
0146 CDB501  >             CALL       OUTMOD        ;OUTPUT THE ETX AGAIN.
0149 C9                    RET                      ;RETURN TO THE CALLING ROUTINE.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                    Sheet 7
XMTON SUBROUTINE

                                                     ;
                                                     ;THIS SUBROUTINE TURNS ON THE AFSATCOM
                                                     ;TRANSMITTER BY SETTING THE RTS LINE
                                                     ;FROM THE USART TO A "1".  THE STATUS
                                                     ;OF THE DTR BIT IS PRESERVED SINCE THE
                                                     ;DTR BIT IS USED AS A FLAG TO INDICATE
                                                     ;WHETHER THE TEST IS REGENERATIVE OR NOT.
                                                     ;
014A F5             XMTON     PUSH      PSW          ;STORE A IN STACK.
014B DBD3                     IN        OD3H         ;READ THE MODEM USART STATUS WORD.
014D E680                     ANI       80H          ;IS THE TEST REGENERATIVE?
014F C25801  >                JNZ       XMTREG       ;JUMP IF REGENERATIVE.
0152 3E35                     MVI       A,035H       ;IF NOT REGEN, LOAD 035H INTO A AND
0154 D3D3                     OUT       OD3H         ;OUTPUT TO MODEM USART CONTROL PORT.
0156 F1                       POP       PSW          ;RESTORE A, AND
0157 C9                       RET                    ;RETURN TO CALLING ROUTINE.
0158 3E37           XMTREG    MVI       A,037H       ;IF REGENERATIVE, LOAD 37H INTO A.
015A C35401  >                JMP       $-6          ;OUTPUT A AND RETURN.
```

Tektronix 8080/8085 ASM V3.3 MESSAGE CONTROLLER VERSION 4
XMT OFF SUBROUTINE

```
                                                  ;
                                                  ;THIS SUBROUTINE WAITS UNTIL THE MODEM USART
                                                  ;TRANSMIT BUFFER IS EMPTY, THEN WAITS FOR THE MODEM
                                                  ;XMT BIT CLOCK TO FALL, INDICATING THAT
                                                  ;THE MODEM HAS SAMPLED THE LAST BIT OF THE
                                                  ;LAST CHARACTER. THE USART CHIP IS THEN RESET.
                                                  ;RESETTING THE CHIP TURNS OFF THE TRANSMITTER
                                                  ;BY SETTING THE RTS BIT EQUAL TO 0.
                                                  ;
015D DBD3        XMTOFF     IN         OD3H       ;READ "SART STATUS.
015F E604                   ANI        04H        ;IS THE TRANSMIT BUFFER EMPTY?
0161 CA5D01   >             JZ         XMTOFF     ;IF NOT EMPTY, READ STATUS AGAIN.
0164 DBEA                   IN         OEAH       ;READ PIO AND
0166 E602                   ANI        02H        ;TEST IF TX CLOCK = 0.
0168 CA6401   >             JZ         $-4        ;IF CLOCK EQUALS ZERO, WAIT.
                                                  ;WHEN CLOCK EQUALS ONE, RESET THE USART.
016B 3E40                   MVI        A,40H      ;SENDING 040H TO PORT
016D D3D3                   OUT        OD3H       ;OD3H RESETS THE USART.
016F 00                     NOP                   ;WAIT
0170 00                     NOP                   ;WAIT
0171 3ECC                   MVI        A,OCCH     ;OUTPUT MODE WORD OCCH
0173 D3D3                   OUT        OD3H       ;TO PORT OD3H.
0175 00                     NOP                   ;WAIT
0176 00                     NOP                   ;WAIT
0177 3E5B                   MVI        A,05BH     ;OUTPUT SYNCH WORD 05BH
0179 D3D3                   OUT        OD3H       ;TO OD3H.
017B 00                     NOP                   ;WAIT
017C 00                     NOP                   ;WAIT
017D 3E94                   MVI        A,94H      ;OUTPUT COMMAND WORD 094H
017F D3D3                   OUT        OD3H       ;TO PORT OD3H.
0181 C9                     RET                   ;RETURN TO CALLING ROUTINE.
```

CONIN SUBROUTINE

```
                                                 ;
                                                 ;THIS SUBROUTINE INPUTS ONE CHARACTER
                                                 ;FROM THE ASR AND RETURNS IT TO THE
                                                 ;CALLING ROUTINE IN THE ACCUMULATOR.
                                                 ;
0182 DBD1          CONIN     IN        OD1H      ;READ THE USART STATUS WORD.
0184 E602                    ANI       02H       ;TEST IF A CHARACTER HAS BEEN TYPED.
0186 CA8201  >               JZ        CONIN     ;IF NOT READ, STATUS AGAIN.
0189 DBD0                    IN        OD0H      ;IF A CHARACTER HAS BEEN TYPED, READ IT
018B D3D0                    OUT       OD0H      ;AND ECHO IT TO THE PRINTER.
018D C9                      RET                 ;RETURN TO CALLING ROUTINE.
```

Tektronix 8080/8085 ASM V3.3 MESSAGE CONTROLLER VERSION 4   Sheet 10
TOGGLE SUBROUTINE

```
                                                     ;
                                                     ;THIS SUBROUTINE IS USED WHEN SIMULATING
                                                     ;A SIGNAL FROM A REGENERATIVE SATELLITE.
                                                     ;BETWEEN MESSAGES, THE TRANSMITTER REMAINS ON
                                                     ;AND A TEN-CHARACTER TABLE IS TRANSMITTED
                                                     ;THE NUMBER OF TIMES INDICATED BY THE VALUE
                                                     ;STORED IN MEMORY LOCATION BFF0H. THIS IS
                                                     ;THE SAME NUMBER USED TO DETERMINE THE NUMBER
                                                     ;OF SECONDS OF DELAY BETWEEN MESSAGES
                                                     ;IN THE NON-REGENERATIVE MODE.
                                                     ;
018E C5               TOGGLE     PUSH      B         ;SAVE THE B-C PAIR ON THE STACK.
018F 3AF0BF                      LDA       0BFF0H    ;MOVE THE NUMBER IN LOCATION BFF0 INTO A.
0192 FE00                        CPI       00H       ;IS IT ZERO?
0194 CAA801     >                JZ        ENDTOG    ;IF SO, DONT TOGGLE.
0197 47                          MOV       B,A       ;SAVE A IN REGISTER B.
0198 216504     >     TOGINIT    LXI       H,TA5     ;POINT HL TO THE TOGGLE TABLE.
019B CDAA01     >                CALL      XMTMSG2   ;TRANSMIT THE TABLE ONCE.
019E 78                          MOV       A,B       ;MOVE THE STORED COUNT TO A.
019F C699                        ADI       099H      ;ADD 99 TO DECREMENT THE COUNT.
01A1 27                          DAA                 ;ADJUST A TO REPRESENT BCD.
01A2 47                          MOV       B,A       ;STORE THE COUNT IN B.
01A3 FE00                        CPI       00H       ;HAS THE COUNT REACHED ZERO?
01A5 C29801     >                JNZ       TOGINIT   ;IF NOT ZERO, SEND TABLE AGAIN.
01A8 C1               ENDTOG     POP       B         ;RESTORE THE BC PAIR.
01A9 C9                          RET                 ;RETURN TO THE CALLING ROUTINE.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 11
XMTMSG2 SUBROUTINE

                                                    ;
                                                    ;THIS SUBROUTINE OUTPUTS A TABLE IN MEMORY
                                                    ;TO THE AFSATCOM MODEM.  PRIOR TO CALLING
                                                    ;THIS SUBROUTINE THE TRANSMITTER MUST BE
                                                    ;TURNED ON AND REGISTER PAIR HL MUST BE
                                                    ;POINTED TO THE FIRST CHARACTER IN THE
                                                    ;TABLE.  CONTROL RETURNS TO THE CALLING
                                                    ;ROUTINE WHEN A 00H END OF FILE MARKER
                                                    ;IS ENCOUNTERED.
                                                    ;
01AA 7E            XMTMSG2   MOV       A,M          ;GET A CHARACTER FROM MEMORY.
01AB FE00                    CPI       00H          ;IS IT THE END OF FILE?
01AD C8                      RZ                     ;IF SO, RETURN TO CALLING ROUTINE.
01AE CDB501  >               CALL      OUTMOD       ;OTHERWISE, OUTPUT THE CHARACTER
                                                    ;TO THE AFSATCOM MODEM.
01B1 23                      INX       H            ;POINT TO THE NEXT CHARACTER.
01B2 C3AA01  >               JMP       XMTMSG2      ;AND REPEAT.

Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 12
OUTMOD SUBROUTINE

                                                    ;
                                                    ;THIS SUBROUTINE OUTPUTS THE CHARACTER IN A
                                                    ;TO THE AFSATCOM MODEM THEN RETURNS TO
                                                    ;THE CALLING ROUTINE.
                                                    ;
01B5 F5            OUTMOD    PUSH      PSW          ;STORE A TEMPORARILY ON THE STACK.
01B6 DBD3                    IN        0D3H         ;READ THE MODEM USART STATUS WORD.
01B8 E601                    ANI       01H          ;IS THE TRANSMITTER READY?
01BA CAB601  >               JZ        $-4          ;IF NOT READY, WAIT.
01BD F1                      POP       PSW          ;WHEN TX READY, POP THE CHARACTER OFF
01BE 2F                      CMA                    ;THE STACK, COMPLIMENT IT, AND
01BF D3D2                    OUT       0D2H         ;OUTPUT IT TO THE MODEM USART DATA PORT.
01C1 C9                      RET                    ;THEN RETURN TO THE CALLING ROUTINE.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                 Sheet 13
MSGNMBR SUBROUTINE

                                                     ;
                                                     ;THIS SUBROUTINE IS CALLED BY THE XMTMSG
                                                     ;SUBROUTINE WHEN A CONTROL Z (SUB) IS
                                                     ;ENCOUNTERED IN THE MESSAGE.  MSGNMBR
                                                     ;TRANSMITS THE MESSAGE COUNT STORED IN
                                                     ;LOCATIONS BFF2H AND BFF3H AND THEN RETURNS
                                                     ;CONTROL TO XMTMSG.  THE MESSAGE COUNT IS
                                                     ;RESET AT THE START OF EACH TEST AND
                                                     ;IS INCREMENTED EACH TIME A MESSAGE HAS
                                                     ;BEEN TRANSMITTED.  ONLY THE THREE LEAST
                                                     ;SIGNIFIGANT DIGITS ARE TRANSMITTED OUT
                                                     ;OF A FOUR-DIGIT DECIMAL MESSAGE COUNT.
                                                     ;
01C2 E5               MSGNMBR   PUSH      H          ;SAVE THE CONTENTS OF THE HL PAIR ON THE STACK.
01C3 21F2BF                     LXI       H,0BFF2H   ;FETCH THE MESSAGE COUNT FROM MEMORY
01C6 46                         MOV       B,M        ;AND MOVE IT TO REGISTER PAIR BC.
01C7 23                         INX       H          ;
01C8 4E                         MOV       C,M        ;
01C9 78                         MOV       A,B        ;MOVE THE TWO MOST SIGNIFIGANT BCD DIGITS TO A.
01CA CDE101  >                  CALL      ASKEY      ;CONVERT THE 4 LSB IN A TO ASCII CODE AND
01CD CDB501  >                  CALL      OUTMOD     ;OUTPUT TO THE MODEM.
01D0 79                         MOV       A,C        ;MOVE THE 2 LEAST SIGNIFIGANT BCD DIGITS TO A.
01D1 0F                         RRC                  ;SHIFT RIGHT
01D2 0F                         RRC                  ;A TOTAL
01D3 0F                         RRC                  ;OF
01D4 0F                         RRC                  ;FOUR PLACES.
01D5 CDE101  >                  CALL      ASKEY      ;CONVERT THE 4 LSB IN A TO ASCII CODE.
01D8 CDB501  >                  CALL      OUTMOD     ;OUTPUT TO MODEM.
01DB 79                         MOV       A,C        ;MOVE TWO LEAST SIGNIFIGANT BCD DIGITS TO A.
01DC CDE101  >                  CALL      ASKEY      ;CONVERT THE LEAST SIGNIFIGANT BCD DIGIT TO ASCII,
01DF E1                         POP       H          ;RESTORE HL, AND
01E0 C9                         RET                  ;RETURN. THE CALLING ROUTINE WILL
                                                     ;OUTPUT THE LAST DIGIT, WHICH IS STILL IN A.
```

ASKEY SUBROUTINE

```
00372                                                      ;
00373                                                      ;THIS SUBROUTINE CONVERTS THE 4 LEAST
00374                                                      ;SIGNIFIGANT DIGITS IN THE ACCUMULATOR
00375                                                      ;TO ASCII CODE, GENERATES ODD PARITY AND
00376                                                      ;RETURNS THE RESULT IN THE ACCUMULATOR TO
00377                                                      ;THE CALLING ROUTINE.
00378                                                      ;THE FOUR MOST SIGNIFIGANT BITS IN
00379                                                      ;THE ACCUMULATOR WHEN ASKEY IS CALLED
00380                                                      ;ARE LOST.
00381                                                      ;
00382 01E1 E60F      ASKEY     ANI        0FH          ;MASK THE 4 MSB.
00383 01E3 C630                ADI        30H          ;CONVERT TO ASCII CODE.
00384 01E5 E0                  RPO                     ;RETURN IF PARITY ODD.
00385 01E6 C680                ADI        080H         ;IF PARITY EVEN, SET PARITY BIT=1.
00386 01E8 C9                  RET                     ;RETURN.
00387                                                      ;
```

PAUSE SUBROUTINE

```
                                                  ;
                                                  ;THIS SUBROUTINE PAUSES FOR THE NUMBER
                                                  ;OF SECONDS EQUAL TO THE NUMBER STORED
                                                  ;AT LOCATION OBFFOH.
                                                  ;
01E9 C5          PAUSE       PUSH     B           ;SAVE BC PAIR ON THE STACK.
01EA 3AFOBF                  LDA      OBFFOH      ;LOAD THE NUMBER AT BFFOH INTO A.
01ED FE00                    CPI      OOH         ;IS IT = 0?
01EF CA0B02  >               JZ       ENDPAUSE    ;IF ZERO, DONT DELAY.
01F2 01CE57      STARTPAUS   LXI      B,57CEH     ;DO THE LOOP 57CEH TIMES.
01F5 E3          WAIT2       XTHL                 ;DELAY
01F6 E3                      XTHL                 ;DELAY
01F7 E3                      XTHL                 ;DELAY
01F8 E3                      XTHL                 ;DELAY
01F9 E3                      XTHL                 ;DELAY
01FA E3                      XTHL                 ;DELAY
01FB 0D                      DCR      C           ;DECREMENT C
01FC C2F501  >               JNZ      WAIT2       ;IF NOT ZERO, DO THE LOOP AGAIN
01FF 05                      DCR      B           ;DECREMENT B
0200 C2F501  >               JNZ      WAIT2       ;IF NOT ZERO, DO THE LOOP AGAIN.
0203 C699                    ADI      99H         ;WHEN THE BC PAIR EQUALS ZERO, DECREMENT A.
0205 27                      DAA                  ;ADJUST A TO CONTAIN TWO BCD DIGITS.
0206 FE00                    CPI      OOH         ;IS A ZERO?
0208 C2F201  >               JNZ      STARTPAUS   ;IF A NOT ZERO, DELAY ANOTHER SECOND.
020B C1          ENDPAUSE    POP      B           ;RESTORE B AND
020C C9                      RET                  ;RETURN TO CALLING ROUTINE.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 16
EPARITY SUBROUTINE

                                                      ;
                                                      ;THIS SUBROUTINE IS CALLED BY THE XMTMSG
                                                      ;SUBROUTINE WHENEVER A CONTROL/ (US)
                                                      ;IS ENCOUNTERED IN THE TRANSMIT MESSAGE.
                                                      ;THIS SUBROUTINE CAUSES THE CHARACTER
                                                      ;IMMEDIATELY FOLLOWING THE US TO BE
                                                      ;TRANSMITTED WITH EVEN PARITY.  THE
                                                      ;US CHARACTER IS NOT TRANSMITTED.
                                                      ;
0200D 23           EPARITY    INX         H           ;POINT H TO THE NEXT CHARACTER IN MEMORY.
020E 7E                       MOV         A,M         ;FETCH THE NEXT CHARACTER INTO A.
020F E67F                     ANI         07FH        ;STRIP OFF THE PARITY BIT.
0211 E8                       RPE                     ;RETURN TO CALLING ROUTINE IF PARITY EVEN.
0212 C680                     ADI         80H         ;IF PARITY ODD, SET THE PARITY BIT
0214 C9                       RET                     ;AND RETURN.


Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 17
REGEN ROUTINE

                                                      ;
                                                      ;THIS ROUTINE IS USED WHEN THE TEST
                                                      ;IS BEGUN BY TYPING R, FOR A REGENERATIVE
                                                      ;SIMULATION. THE DTR BIT IS SET = 1, AND THE
                                                      ;FRONT PANEL LIGHT LABELED "REGEN" IS
                                                      ;ILLUMINATED.
                                                      ;
0215 3E17          REGEN      MVI         A,17H       ;OUTPUT CONTROL WORD 17H TO
0217 D3D3                     OUT         0D3H        ;USART CONTROL PORT 0D3H.
0219 C39400  >                JMP         TESTCON     ;
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 18
NONREGEN

                                                      ;
                                                      ;THIS ROUTINE IS USED WHEN THE TEST IS BEGUN
                                                      ;BY TYPING N, TO SIGNIFY A NON-REGENERATIVE
                                                      ;TEST.  THE DTR BIT IS SET = 0, AND THE
                                                      ;FRONT PANEL "REGEN" LIGHT IS EXTINGUISHED.
                                                      ;
021C 3E15        NONREGEN   MVI       A,15H           ;OUTPUT CONTROL WORD 15H TO
021E D3D3                   OUT       OD3H            ;USART CONTROL PORT OD3H.
0220 C39400  >              JMP       TESTCON         ;RETURN.

Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 19
PRTMSG SUBROUTINE

                                                      ;
                                                      ;THIS SUBROUTINE PRINTS A MESSAGE FROM MEMORY
                                                      ;ON THE SYSTEM CONSOLE.  PRIOR TO CALLING
                                                      ;PRTMSG, THE HL PAIR MUST BE POINTED TO THE
                                                      ;ADDRESS OF THE FIRST CHARACTER OF THE MESSAGE.
                                                      ;CONTROL IS RETURNED TO THE CALLING ROUTINE
                                                      ;WHEN A OOH END OF FILE MARKER IS ENCOUNTERED.
                                                      ;
0223 F5          PRTMSG     PUSH      PSW             ;STORE THE CONTENTS OF A IN STACK.
0224 DBD1        PRTTEST    IN        OD1H            ;READ PRINTER USART STATUS WORD.
0226 E601                   ANI       01H             ;IS USART READY FOR A CHARACTER?
0228 CA2402  >              JZ        PRTTEST         ;IF NOT, WAIT UNTIL IT IS READY.
022B 7E                     MOV       A,M             ;MOVE A CHARACTER INTO A.
022C FE00                   CPI       OOH             ;IS IT AN END OF FILE MARKER?
022E CA3702  >              JZ        ENDPRINT        ;JUMP IF END OF FILE.
0231 D3D0                   OUT       ODOH            ;IF NOT END OF FILE, OUTPUT THE CHARACTER,
0233 23                     INX       H               ;POINT THE HL PAIR TO THE NEXT CHARACER,
0234 C32402  >              JMP       PRTTEST         ;AND PREPARE TO PRINT THE NEXT CHARACTER.
0237 F1          ENDPRINT   POP       PSW             ;RESTORE A AND
0238 C9                     RET                       ;RETURN TO THE CALLING ROUTINE.
```

```
                                                ;THIS SUBROUTINE IS CALLED BY THE XMTMSG
                                                ;SUBROUTINE WHEN A CONTROL COMMA (FS)
                                                ;CHARACTER IS ENCOUNTERED IN THE MESSAGE.
                                                ;THE CONTROL COMMA IS NOT TRANSMITTED.
                                                ;INSTEAD, THE NEXT TWO CHARACTERS OF THE
                                                ;MESSAGE ARE READ, CONVERTED FROM ASCII
                                                ;TO PACKED BCD, AND THE RESULTING VALUE USED
                                                ;TO DETERMINE THE NUMBER OF SECONDS OF
                                                ;RANDOM DATA WHICH WHICH WILL BE TRANSMITTED
                                                ;VIA THE AFSATCOM MODEM BEFORE CONTROL IS RE-
                                                ;TURNED TO THE XMTMSG SUBROUTINE.
0239 C5             TOGGLE2   PUSH    B         ;SAVE BC ON STACK.
023A 23                       INX     H         ;GET THE NEXT CHARACTER
023B 7E                       MOV     A,M       ;AFTER THE CONTROL COMMA.
023C E60F                     ANI     0FH       ;STRIP OFF THE 4 MSB.
023E 0F                       RRC               ;SHIFT A
023F 0F                       RRC               ;TOTAL OF
0240 0F                       RRC               ;FOUR TIMES.
0241 0F                       RRC               ;
0242 47                       MOV     B,A       ;STORE THE VALUE IN B.
0243 23                       INX     H         ;GET THE NEXT CHARACTER OF THE MESSAGE.
0244 7E                       MOV     A,M       ;
0245 E5                       PUSH    H         ;SAVE THE HL PAIR ON STACK.
0246 E60F                     ANI     0FH       ;STRIP OFF THE 4 MSB.
0248 B0                       ORA     B         ;OR THE RESULT WITH B TO GIVE TWO DIGITS OF PACKED BCD.
0249 47                       MOV     B,A       ;STORE THIS IN REGISTER B.
024A FE00                     CPI     00H       ;TEST THE VALUE FOR 00H.
024C CA6202  >                JZ      TOGEND    ;DONT TOGGLE IF B CONTAINS ZERO.
024F 216504  >      LOOP      LXI     H,TA5     ;POINT HL TO TOGGLE TABLE, AND
0252 CDAA01  >                CALL    XMTMSG2   ;TRANSMIT THE CONTENTS OF THE TABLE.
0255 78                       MOV     A,B       ;MOVE THE VALUE IN B TO A.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 21
TOGGLE2 SUBROUTINE

0256 C699                  ADI      99H        ;DECREMENT A.
0258 27                    DAA                 ;ADJUST A TO REPRESENT TWO BCD DIGITS.
0259 47                    MOV      B,A        ;STORE THE NEW VALUE IN B.
025A FE00                  CPI      00H        ;IS THE NEW VALUE = 0?
025C CA6202  >             JZ       TOGEND     ;IF SO, RESTORE REGISTERS AND END.
025F C34F02  >             JMP      LOOP       ;IF NOT, SEND THE TABLE AGAIN.
0262 E1         TOGEND     POP      H          ;RESTORE HL,
0263 C1                    POP      B          ;RESTORE BC,
0264 23                    INX      H          ;POINT HL TO NEXT CHARACTER OF THE MESSAGE,
0265 7E                    MOV      A,M        ;MOVE THE CHARACTER INTO THE ACCUMULATOR,
0266 C9                    RET                 ;AND RETURN.  THE CALLING ROUTINE WILL OUT-
                                               ;PUT THE CHARACTER.
```

```
                                                ;THIS SUBROUTINE PROMPTS THE OPERATOR TO ENTER
                                                ;THE TEST MESSAGE, POINTS THE HL PAIR TO
                                                ;LOCATION 8000H, AND INPUTS AND STORES A MESSAGE
                                                ;TYPED BY THE OPERATOR.  THE CONTROL CHARACTERS
                                                ;ETX, EOT, BS, AND RS ARE RECOGNIZED. CONTROL
                                                ;RETURNS TO THE CALLING ROUTINE WHEN AN ETX OR
                                                ;EOT IS TYPED.  BS (BACKSPACE) ALLOWS THE OPER-
                                                ;ATOR TO TYPE OVER AN INCORRECT CHARACTER, AND
                                                ;RS ALLOWS THE OPERATOR TO BEGIN ENTERING THE
                                                ;MESSAGE AGAIN.
                                                ;
0267 217004  >  MSGIN     LXI     H,TA6         ;POINT HL TO THE PROMPT
026A CD2302  >            CALL    PRTMSG        ;PRINT: "ENTER TEST MESSAGE, END WITH ETX OR EOT."
026D 210080               LXI     H,8000H       ;POINT HL TO THE BUFFER SPACE AND
0270 CD8201  >  INPUT     CALL    CONIN         ;INPUT 1 CHARACTER.
0273 FE83                 CPI     083H          ;IS IT ETX?
0275 CA9002  >            JZ      ENDMSGIN      ;IF ETX, END THE FILE.
0278 FE04                 CPI     04H           ;IS IT EOT?
027A CA9002  >            JZ      ENDMSGIN      ;IF EOT, END THE FILE.
027D FE9E                 CPI     09EH          ;IS IT RS?
027F CA6702  >            JZ      MSGIN         ;IF RS, BEGIN AGAIN.
0282 FE08                 CPI     08H           ;IS IT A BACKSPACE?
0284 C28B02  >            JNZ     STORE         ;IF NONE OF THE ABOVE, STORE THE CHARACTER.
0287 2B                   DCX     H             ;IF BACKSPACE, DECREMENT H,
0288 C37002  >            JMP     INPUT         ;AND GET THE NEXT CHARACTER.
028B 77         STORE     MOV     M,A           ;STORE THE CHAR IN A,
028C 23                   INX     H             ;INCREMENT H,
028D C37002  >            JMP     INPUT         ;AND INPUT ANOTHER CHARACTER.
0290 77         ENDMSGIN  MOV     M,A           ;STORE THE ETX OR EOT.
0291 AF                   XRA     A             ;CLEAR THE ACCUMULATOR, AND
0292 23                   INX     H             ;STORE THE 00 IN THE NEXT MEMORY LOCATION.
0293 77                   MOV     M,A           ;
0294 C9                   RET                   ;RETURN.
```

```
                                                       ;THIS SUBROUTINE FIRST ASKS THE OPERATOR HOW
                                                       ;MANY TIMES THE MESSAGE IS TO BE TRANSMITTED.
                                                       ;THEN IT INPUTS FOUR DIGITS FROM THE CONSOLE
                                                       ;AND STORES THEM, IN PACKED BCD FORM, IN
                                                       ;REGISTER PAIR DE.  IT THEN ASKS THE OPERATOR
                                                       ;HOW MUCH DELAY IS TO BE PUT BETWEEN MESSAGES,
                                                       ;AND INPUTS 2 CHARACTERS WHICH WILL BE STORED IN
                                                       ;MEMORY LOCATION BFFOH AS PACKED BCD.  IF
                                                       ;FEWER THAN THE ALLOTTED NUMBER OF CHARACTERS
                                                       ;ARE TYPED, IT IS ASSUMED THAT THE OPERATOR
                                                       ;DIDNT TYPE THE LEADING ZEROES.  IF MORE THAN
                                                       ;FOUR DIGITS ARE TYPED PRIOR TO TYPING
                                                       ;"RETURN", THE NUMBERS TYPED FIRST ARE DISRE-
                                                       ;GARDED, GIVING THE OPERATOR AN OPPORTUNITY
                                                       ;TO CORRECT ERRORS.   FOR BOTH PARAMETERS,
                                                       ;INPUT CEASES WHEN CARRIAGE RETURN IS TYPED.
                                                       ;IF A CHARACTER OTHER THAN 0-9 IS TYPED,
                                                       ;THE COMPUTER REQUESTS THE OPERATOR TO
                                                       ;BEGIN ANEW.
                                                       ;
0295 21AC04  >  MSGCOUNT    LXI        H,TA7           ;POINT HL TO PROMPT.
0298 CD2302  >              CALL       PRTMSG          ;PRINT: "ENTER THE NUMBER OF TIMES THE
                                                       ;MESSAGE IS TO BE SENT."
029B 110000                 LXI        D,0000H         ;CLEAR THE DE PAIR.
029E CDE702  >  NEXT        CALL       BCDIN           ;INPUT 1 CHARACTER AND CONVERT TO BCD.
02A1 FEFF                   CPI        0FFH            ;IF FFH IS RETURNED BY BCD SUBROUTINE,
02A3 CA9502  >              JZ         MSGCOUNT        ;A NON-VALID CHARACTER WAS TYPED,
                                                       ;TRY AGAIN.
02A6 FE0D                   CPI        0DH             ;WAS A CARRIAGE RETURN TYPED?
02A8 CAC002  >              JZ         DELAYIN         ;IF SO, ENTER THE NEXT PARAMETER.
02AB 47                     MOV        B,A             ;STORE THE CHARACTER TEMPORARILY IN B.
02AC 0E04                   MVI        C,04H           ;SET THE COUNTER TO 4.
02AE 7B         SHIFT4      MOV        A,E             ;SHIFT THE DE PAIR FOUR PLACES TO THE LEFT BY:
```

```
02AF 17                          RAL                  ;ROTATING A 1 PLACE LEFT THROUGH THE CARRY BIT,
02B0 5F                          MOV        E,A       ;PUTTING THE SHIFTED VALUE BACK IN E,
02B1 7A                          MOV        A,D       ;MOVING D TO ACCUMULATOR,
02B2 17                          RAL                  ;SHIFTING ONE PLACE LEFT THROUGH THE CARRY BIT,
02B3 57                          MOV        D,A       ;AND PUTTING THE RESULT IN D.
02B4 0D                          DCR        C         ;HAS THIS BEEN DONE FOUR TIMES?
02B5 C2AE02  >                   JNZ        SHIFT4    ;IF NOT, DO IT AGAIN.
02B8 7B                          MOV        A,E       ;MOVE THE NUMBER IN E TO A.
02B9 E6F0                        ANI        0F0H      ;MASK THE RIGHT FOUR BITS OF A.
02BB B0                          ORA        B         ;PLACE THE MOST RECENTLY INPUTTED BCD DIGIT INTO
02BC 5F                          MOV        E,A       ;THE 4 MSB POSITIONS OF A AND PUT THE RESULT
02BD C39E02  >                   JMP        NEXT      ;BACK IN E.  INPUT ANOTHER CHARACTER.
02C0 210E05  >  DELAYIN          LXI        H,TA8     ;POINT HL TO PROMPT.
02C3 CD2302  >                   CALL       PRTMSG    ;PRINT: "ENTER THE NUMBER OF SECONDS OF DELAY---"
02C6 0E00                        MVI        C,00H     ;CLEAR REGISTER C.
02C8 CDE702  >  INPUT2           CALL       BCDIN     ;INPUT 1 BCD CHARACTER
02CB FEFF                        CPI        0FFH      ;WAS IT A VALID CHARACTER BETWEEN 0 AND 9?
02CD CAC002  >                   JZ         DELAYIN   ;IF NOT, PROMPT OPERATOR AND TRY AGAIN.
02D0 FE0D                        CPI        0DH       ;WAS IT A CARRIAGE RETURN?
02D2 CAE202  >                   JZ         ENDDELAY  ;
02D5 47                          MOV        B,A       ;STORE THE BCD VALUE IN B TEMPORARILY
02D6 79                          MOV        A,C       ;
02D7 07                          RLC                  ;ROTATE THE VALUE FROM C
02D8 07                          RLC                  ;FOUR PLACES TO THE LEFT.
02D9 07                          RLC                  ;
02DA 07                          RLC                  ;
02DB E6F0                        ANI        0F0H      ;CLEAR THE 4 LSB.
02DD B0                          ORA        B         ;PUT THE BCD VALUE IN B INTO THE 4 LSB POSITIONS
02DE 4F                          MOV        C,A       ;AND PUT THE RESULT BACK IN C.
02DF C3C802  >                   JMP        INPUT2    ;GET THE NEXT CHARACTER.
02E2 21F0BF     ENDDELAY         LXI        H,0BFF0H  ;STORE REGISTER C IN LOC BFF0H.
02E5 71                          MOV        M,C       ;
02E6 C9                          RET                  ;RETURN
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 25
BCDIN SUBROUTINE

                                                   ;
                                                   ;THIS SUBROUTINE INPUTS ONE CHARACTER FROM THE
                                                   ;CONSOLE AND TESTS IT.  ANY VALID ASCII
                                                   ;CHARACTER REPRESENTING A NUMBER BETWEEN
                                                   ;0 AND 9 IS CONVERTED TO BCD AND RETURNED TO
                                                   ;THE CALLING ROUTINE IN THE FOUR MSB POSITIONS
                                                   ;OF THE ACCUMULATOR.  A CARRIAGE RETURN IS
                                                   ;RETURNED TO THE CALLING ROUTINE UNALTERED.
                                                   ;ANY OTHER CHARACTER CAUSES OFFH TO BE
                                                   ;RETURNED IN A.
                                                   ;
02E7 CD8201  >  BCDIN      CALL       CONIN        ;INPUT 1 CHARACTER.
02EA E67F                  ANI        07FH         ;STRIP THE PARITY BIT.
02EC FE0D                  CPI        0DH          ;IS IT A CARRIAGE RETURN?
02EE C8                    RZ                      ;IF SO, RETURN TO THE CALLING ROUTINE.
02EF FE30                  CPI        030H         ;COMPARE WITH ASCII ZERO.
02F1 FAFF02  >             JM         ENDBCD       ;IF LESS THAN 30H, THE CHARACTER IS INVALID.
02F4 FE3A                  CPI        03AH         ;COMPARE WITH ASCII NINE+1.
02F6 F2FF02  >             JP         ENDBCD       ;IF EQUQL TO QR GREATER THAN 03AH,
02F9 CAFF02  >             JZ         ENDBCD       ;THE CHARACTER IS INVALID.
02FC E60F                  ANI        0FH          ;IF A VALID CHARACTER, CLEAR THE 4 MSB.
02FE C9                    RET                     ;RETURN.
02FF 3EFF       ENDBCD     MVI        A,0FFH       ;TELL THE CALLING ROUTINE AN INVALID
0301 C9                    RET                     ;CHARACTER WAS TYPED.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                    Sheet 26
CLEARCOUNT SUBROUTINE

                                                 ;
                                                 ;THIS SUBROUTINE RESETS THE MESSAGE COUNT AT
                                                 ;LOCATIONS OBFF2H AND OBFF3H.
                                                 ;
0302 AF        CLEARCOUNT XRA      A             ;CLEAR A.
0303 21F2BF               LXI      H,OBFF2H      ;POINT H TO BFF2H.
0306 77                   MOV      M,A           ;STORE OOH IN LOC. OBFF2H.
0307 23                   INX      H             ;
0308 77                   MOV      M,A           ;STORE OOH IN LCC. OBFF3H.
0309 C9                   RET                    ;RETURN.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                    Sheet 27
AUXILIARY ROUTINES INCLUDED FOR EASE OF FUTURE EXPANSION

                                                 ;
                                                 ;FOR CONVENIENCE OF FUTURE EXPANSION, FOUR UTILTIY
                                                 ;SUBROUTINES ARE INCLUDED WHICH ARE NOT USED AT THE
                                                 ;PRESENT TIME.  THESE ARE (1) PORT2OUT,(2) PORT3OUT,
                                                 ;(3)CHARIN2, ANr (4) CHARIN3.
                                                 ;
                                                 ;DURING THE INITIALIZATION, SERIAL PORTS 2 AND 3
                                                 ;ARE INITIALIZED TO 300 BAUD ASYNCHRONOUS
                                                 ;OPERATION, WITH EVEN PARITY.  IF OPERATION AT
                                                 ;ANOTHER SPEED IS DESIRED, THE USARTS CAN BE
                                                 ;REINITIALIZED TO ANY SPEED BY THE USER WRITTEN
                                                 ;PROGRAM STARTING AT LOCATION 0800H.  THE
                                                 ;SUBROUTINES INCLUDED HERE WILL RUN REGARDLESS.
                                                 ;OF THE BAUD RATE SELECTED.
                                                 ;
                                                 ;THESE FOUR UTILITY SUBROUTINES ARE LISTED ON
                                                 ;THE NEXT FOUR PAGES.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                   Sheet 28
PORT2OUT SUBROU.INE

                                                      ;
                                                      ;THIS SUBROUTINE OUTPUTS A TABLE FROM MEMORY
                                                      ;VIA USART 2.  PRIOR TO CALLING THIS SUBROUTINE,
                                                      ;THE HL PAIR MUST BE POINTED TO THE LOCATION
                                                      ;OF THE FIRST CHARACTER OF THE TABLE TO BE
                                                      ;OUTPUTTED.
                                                      ;CONTROL IS RETURNED TO THE CALLING ROUTINE WHEN
                                                      ;A 00H END OF FILE MARKER IS ENCOUNTERED.
                                                      ;
030A DBD5        PORT2OUT   IN        0D5H            ;READ USART 2 STATUS.
030C E601                   ANI       01H             ;IS TX READY?
030E CA0A03  >              JZ        PORT2OUT        ;IF NOT, TRY AGAIN.
0311 7E                     MOV       A,M             ;WHEN TX IS READY, GET A CHARACTER FROM MEMORY
0312 FE00                   CPI       00H             ;AND TEST IT FOR END OF FILE.
0314 C8                     RZ                        ;IF END OF FILE, RETURN.
0315 D3D4                   OUT       0D4H            ;IF NOT END OF FILE, OUTPUT THE CHARACTER.
0317 23                     INX       H               ;POINT TO THE NEXT CHARACTER IN THE TABLE.
0318 C30A03  >              JMP       PORT2OUT        ;REPEAT THE PROCESS.

Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                   Sheet 29
PORT3OUT SUBROUTINE

                                                      ;
                                                      ;THIS SUBROUTINE IS EXACTLY THE SAME AS
                                                      ;PORT2OUT, EXCEPT THAT IT OUTPUTS VIA
                                                      ;SERIAL PORT 3.
                                                      ;
031B DBD7        PORT3OUT   IN        0D7H            ;READ USART 3 STATUS WORD.
031D E601                   ANI       01H             ;IS TX READY?
031F CA1B03  >              JZ        PORT3OUT        ;IF TX NOT READY, TRY AGAIN.
0322 7E                     MOV       A,M             ;GET A CHARACTER FROM MEMORY
0323 FE00                   CPI       00H             ;AND TEST IT FOR END OF FILE.
0325 C8                     RZ                        ;RETURN IF END OF FILE.
0326 D3D6                   OUT       0D6H            ;IF NOT END OF FILE, OUTPUT THE CHARACTER.
0328 23                     INX       H               ;POINT HL TO THE NEXT CHARACTER IN THE TABLE
0329 C31B03  >              JMP       PORT3OUT        ;REPEAT THE PROCESS.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 30
CHARIN2 SUBROUTINE

00723                                                   ;
00724                                                   ;THIS SUBROUTINE INPUTS ONE CHARACTER VIA
00725                                                   ;SERIAL PORT 2.
00726                                                   ;
00727 032C DBD5        CHARIN2    IN          0D5H      ;READ USART 2 STATUS.
00728 032E E602                   ANI         02H       ;TEST FOR RCVR READY.
00729 0330 CA2C03  >              JZ          CHARIN2   ;IF NOT READY, TRY AGAIN.
00730 0333 DBD4                   IN          0D4H      ;WHEN RCV READY, INPUT A CHARACTER
00731 0335 C9                     RET                   ;AND RETURN.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 31
CHARIN3 SUBROUTINE

00734                                                   ;
00735                                                   ;THIS SUBROUTINE INPUTS ONE CHARACTER VIA
00736                                                   ;SERIAL I/O PORT 3.
00737                                                   ;
00738 0336 DBD7        CHARIN3    IN          0D7H      ;READ USART 3 STATUS.
00739 0338 E602                   ANI         02H       ;TEST FOR RCVR READY.
00740 033A CA3603  >              JZ          CHARIN3   ;IF NOT READY, TRY AGAIN.
00741 033D DBD6                   IN          0D6H      ;INPUT 1 BYTE AND
00742 033F C9                     RET                   ;RETURN TO CALLING ROUTINE.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                  Sheet 32
RCVMSG SUBROUTINE

                                                    ;THIS SUBROUTINE INPUTS A MESSAGE FROM THE
                                                    ;AFSATCOM MODEM TO THE COMPUTER MEMORY.
                                                    ;PRIOR TO CALLING THIS ROUTINE IT IS NECESSARY
                                                    ;TO POINT THE HL PAIR TO THE ADDRESS IN
                                                    ;MEMORY WHERE THE FIRST RECEIVED CHARACTER
                                                    ;IS TO BE STORED.  PRIOR TO CALLING RCVMSG,
                                                    ;IT IS ALSO NECESSARY TO DETERMINE THAT THE
                                                    ;MODEM IS ACTUALLY RECEIVING A MESSAGE.
                                                    ;
0340 DBD3      RCVMSG     IN         OD3H          ;READ THE USART STATUS BIT
0342 E602                 ANI        02H           ;AND TEST IT. IS RCVR READY?
0344 CA5603  >            JZ         TSTCLK        ;IF RCV NOT READY, IS BIT CLOCK RUNNING?
0347 DBD2                 IN         OD2H          ;WHEN RCVR READY: INPUT A CHARACTER,
0349 2F                   CMA                      ;COMPLIMENT IT,
034A 77                   MOV        M,A           ;AND STORE IT.
034B FE83                 CPI        083H          ;IS IT ETX?
034D CA6A03  >            JZ         PRNTETX       ;IF SO PRINT ETX ON PRINTER.
0350 D3D0                 OUT        ODOH          ;IF NOT ETX, PRINT IT.
0352 23                   INX        H             ;THEN POINT TO THE NEXT ADDRESS AND
0353 C34003  >            JMP        RCVMSG        ;INPUT THE NEXT CHARACTER.
0356 DBEB      TSTCLK     IN         OEBH          ;TEST IF BIT CLOCK IS RUNNING.
0358 E602                 ANI        02H           ;
035A C24003  >            JNZ        RCVMSG        ;IF BIT CLOCK IS STILL RUNNING,
                                                    ;TEST USART TO SEE IF CHARACTER HAS BEEN RECEIVED.
035D 3600                 MVI        M,00H         ;WHEN BIT CLOCK STOPS, PUT A 00H END OF
                                                    ;FILE MARKER AT THE END OF THE MESSAGE IN
                                                    ;MEMORY.
035F 3E94                 MVI        A,94H         ;OUTPUT 94H AS A COMMAND TO THE USART
0361 D3D3                 OUT        OD3H          ;TO ENTER THE HUNT MODE.
0363 217005  >            LXI        H,TA9         ;PRINT A CR LF.
0366 CD2302  >            CALL       PRTMSG        ;
0369 C9                   RET                      ;RETURN TO CALLING ROUTINE.
```

```
RCVMSG SUBROUTINE

                                                         ;
      036A E5           PRNTETX     PUSH      H          ;SAVE THE HL PAIR ON THE STACK.
      036B 217305  >                LXI       H,TA10     ;
      036E CD2302  >                CALL      PRTMSG     ;PRINT ETX ON THE PRINTER.
      0371 E1                       POP       H          ;RESTORE THE HL PAIR.
      0372 23                       INX       H          ;
      0373 C34003  >                JMP       RCVMSG     ;LOOK FOR THE NEXT CHARACTER.
```

```
Tektronix  8080/8085 ASM V3.3  MESSAGE CONTROLLER VERSION 4                    Sheet 34
ASRSIM ROUTINE

00787                                                   ;THIS ROUTINE ALLOWS THE MESSAGE CONTROLLER
00788                                                   ;TO BE USED FOR TWO-WAY COMMUNICATIONS IN
00789                                                   ;A MANNER SOMEWHAT LIKE THE AUTO TRANSMIT MODE
00790                                                   ;OF THE AFSATCOM ASR.  NO MANUAL TRANSMIT OR
00791                                                   ;POLL TRANSMIT IS PROVIDED.  FURTHERMORE,
00792                                                   ;OPERATION IS HALF-DUPLEX ONLY.
00793                                                   ;
00794 0376 DBD1       ASRSIM    IN        0D1H          ;WAS A CHARACTER TYPED?
00795 0378 E602                 ANI       02H           ;
00796 037A CAA003  >            JZ        RCVTEST       ;IF NOT, TEST THE RECEIVER.
00797 037D DBD0                 IN        0D0H          ;IF CHARACTER WAS TYPED, GET THE
00798 037F D3D0                 OUT       0D0H          ;CHARACTER AND ECHO IT.
00799 0381 FE9B                 CPI       09BH          ;IS IT ESC?
00800 0383 CA9103  >            JZ        TRANSMIT      ;IF ESC, SEND THE MESSAGE IN BUFFER.
00801 0386 210080               LXI       H,8000H       ;IF NOT ESC, INPUT A MESSAGE.
00802 0389 77                   MOV       M,A           ;
00803 038A 23                   INX       H             ;
00804 038B CD7002  >            CALL      INPUT         ;
00805 038E C37603  >            JMP       ASRSIM        ;THEN LOOK FOR RECEIVED MESSAGE OR TYPED CHARACTER.
00806 0391 210080     TRANSMIT  LXI       H,8000H       ;POINT HL TO THE BEGINNING OF THE BUFFER.
00807 0394 CD4A01  >            CALL      XMTON         ;TURN ON THE TRANSMITTER.
00808 0397 CD1D01  >            CALL      XMTMSG        ;SEND THE MESSAGE.
00809 039A CD5D01  >            CALL      XMTOFF        ;TURN OFF THE TRANSMITTER
00810 039D C37603  >            JMP       ASRSIM        ;AND LOOK FOR RECEIVED MESSAGE OR TYPED CHARACTER.
00811 03A0 DBEB       RCVTEST   IN        0EBH          ;IS BIT CLOCK RUNNING?
00812 03A2 E602                 ANI       02H           ;
00813 03A4 CA7603  >            JZ        ASRSIM        ;IF NOT, TEST THE ASR FOR TYPED CHARACTERS.
00814 03A7 217005  >            LXI       H,TA9         ;IF BIT CLOCK RUNNING, SEND CR LF TO PRINTER.
00815 03AA CD2302  >            CALL      PRTMSG        ;
00816 03AD 2100A0               LXI       H,0A000H      ;THEN INPUT A MESSAGE FROM DUAL MODEM.
00817 03B0 CD4003  >            CALL      RCVMSG        ;
00818 03B3 C37603  >            JMP       ASRSIM        ;THEN LOOK FOR TYPED CHARACTERS.
```

```
RESERVE BLOCKS OF MEMORY FOR TABLES

00821 03B6 004B       TA1        BLOCK      75          ;RESERVE 75 BYTES FOR TABLE 1. (REPETITIVE MESSAGE.....)
00822 0401 005A       TA2        BLOCK      90          ;RESERVE 90 BYTES FOR TABLE 2. (TYPE R FOR REGEN.....)
00823 045B 0005       TA3        BLOCK      5           ;RESERVE  5 BYTES FOR TABLE 3. (PREAMBLE TABLE)
00824 0460 0005       TA4        BLOCK      5           ;RESERVE  5 BYTES FOR TABLE 4. (POSTAMBLE TABLE)
00825 0465 000B       TA5        BLOCK      11          ;RESERVE 11 BYTES FOR TABLE 5. (TOGGLE)
00826 0470 003C       TA6        BLOCK      60          ;RESERVE 60 BYTES FOR TABLE 6. (ENTER TEST MESSAGE..)
00827 04AC 0062       TA7        BLOCK      98          ;RESERVE 98 BYTES FOR TABLE 7. (ENTER NO. OF MSGS....)
00828 050E 0062       TA8        BLOCK      98          ;RESERVE 98 BYTES FOR TABLE 8. (ENTER NO. OF SECONDS..)
00829 0570 0003       TA9        BLOCK      3           ;RESERVE  3 BYTES FOR TABLE 9. (CR LF)
00830 0573 0005       TA10       BLOCK      5           ;RESERVE  5 BYTES FOR TABLE 10. (ETX ETX)
00831                            END                    ;END OF PROGRAM.
```

```
Tektronix  8080/8085 ASM V3.3  Symbol Table                    Sheet 36

Scalars

    A ------ 0007           B ------ 0000           C ------ 0001
    H ------ 0004           L ------ 0005           M ------ 0006

%TEMPO (default) Section (0578)

    ASKEY -- 01E1           ASRSIM - 0376           BCDIN -- 02E7
    CHARIN3  0336           CLEAR -- 0076           CLEARCOU 0302
    ECHO --- 009D           ENDBCD - 02FF           ENDDELAY 02E2
    ENDPRINT 0237           ENDTOG - 01A8           EPARITY  020D
    INPUT -- 0270           INPUT2 - 02C8           INPUTMSG 0070
    MSGIN -- 0267           MSGNMBR  01C2           NEXT --- 029E
    PAUSE -- 01E9           PORT2OUT 030A           PORT3OUT 031B
    PRTMSG - 0223           PRTTEST  0224           RCVMSG - 0340
    REGEN -- 0215           RESTART  004D           SEND2 -- 00DB
    STARTPAU 01F2           STORE -- 028B           TA1 ---- 03B6
    TA3 ---- 045B           TA4 ---- 0460           TA5 ---- 0465
    TA8 ---- 050E           TA9 ---- 0570           TESTCON  0094
    TOGGLE - 018E           TOGGLE2  0239           TOGINIT  0198
    WAIT --- 00AE           WAIT2 -- 01F5           WHATNEXT 00BA
    XMTOFF - 015D           XMTON -- 014A           XMTREG - 0158


                D ------ 0002           E ------ 0003
                PSW ---- 0006           SP ----- 0006


                BEGIN -- 0050           CHARIN2  032C
                CONIN -- 0182           DELAYIN  02C0
                ENDMSGIN 0290           ENDPAUSE 020B
                ETX ---- 0142           INCREMEN 00EE
                LOOP --- 024F           MSGCOUNT 0295
                NONREGEN 021C           OUTMOD - 01B5
                POSTAMBL 0111           PRNTETX  036A
                RCVTEST  03A0           READY -- 0079
                SENDMSG  00C8           SHIFT4 - 02AE
                TA10 --- 0573           TA2 ---- 0401
                TA6 ---- 0470           TA7 ---- 04AC
                TESTCOUN 0100           TOGEND - 0262
                TRANSMIT 0391           TSTCLK - 0356
                XMTMSG - 011D           XMTMSG2  01AA

    831 Source Lines     831 Assembled Lines    46344 Bytes available

                  >>> No assembly errors detected <<<
```

APPENDIX B

MEMORY DUMP

```
0000=3E 41 D3 E8 3E C0 D3 E9 3E 36 D3 DB D3 DF 3E B6   >A..>...>6....>.
0010=D3 DB 3E 40 D3 D8 D3 DA D3 DC 3E 00 D3 D8 D3 DA   ..>@......>.....
0020=D3 DC 3E 4F D3 D1 3E 7B D3 D5 D3 D7 3E 37 D3 D1   ..>O..>.....>7..
0030=D3 D5 D3 D7 3E CC D3 D3 00 00 3E 5B D3 D3 00 00   ....>.....>[....
0040=3E 94 D3 D3 00 00 00 31 EF BF C3 50 00 CD 5D 01   >......1...P..].
0050=21 B6 03 CD 23 02 CD 82 01 4F 21 70 05 CD 23 02   !...#....O!...#.
0060=79 FE D3 CA 00 08 FE C1 CA 76 03 FE 52 C2 50 00   ............R.P.
0070=CD 67 02 CD 95 02 CD 02 03 21 01 04 CD 23 02 CD   .........!...#..
0080=82 01 FE 52 CA 15 02 FE CE CA 1C 02 FE C1 CA 4D   ...R...........M
0090=00 C2 79 00 DB D1 E6 02 CA C8 00 DB D0 D3 D0 FE   ................
00A0=52 CA 76 00 FE C1 CA 4D 00 FE 20 C2 C8 00 DB D3   R......M.. .....
00B0=E6 80 CA BA 00 3E 16 CD B5 01 DB D1 E6 02 CA AE   .....>..........
00C0=00 DB D0 FE 20 C2 9D 00 CD 4A 01 DB D3 E6 80 CA   .... ....J......
00D0=DB 00 CD 8E 01 21 5B 04 CD AA 01 21 00 80 CD 1D   .....![....!....
00E0=01 DB D3 E6 80 C2 11 01 CD 5D 01 CD E9 01 21 F3   .........]....!.
00F0=BF 7E C6 01 27 77 D2 00 01 2B 7E C6 01 27 77 23   ....'....+...'.#
0100=7E BB C2 94 00 2B 7E BA C2 94 00 CD 5D 01 C3 76   .....+......]...
0110=00 21 60 04 CD AA 01 CD 8E 01 C3 EE 00 7E FE 1A   .!`.............
0120=CC C2 01 FE 1F CC 0D 02 FE 1C CC 39 02 FE 04 C8   ...........9....
0130=FE 83 CA 42 01 00 00 00 00 00 00 CD B5 01 23 C3   ...B..........#.
0140=1D 01 CD B5 01 2F CD B5 01 C9 F5 DB D3 E6 80 C2   ...../..........
0150=58 01 3E 35 D3 D3 F1 C9 3E 37 C3 54 01 DB D3 E6   X.>5....>7.T....
0160=04 CA 5D 01 DB EA E6 02 CA 64 01 3E 40 D3 D3 00   ..]........>@...
0170=00 3E CC D3 D3 00 00 3E 5B D3 D3 00 00 3E 94 D3   .>.....>[....>..
0180=D3 C9 DB D1 E6 02 CA 82 01 DB D0 D3 D0 C9 C5 3A   ...............:
0190=F0 BF FE 00 CA A8 01 47 21 65 04 CD AA 01 78 C6   .......G!.......
01A0=99 27 47 FE 00 C2 98 01 C1 C9 7E FE 00 C8 CD B5   .'G.............
01B0=01 23 C3 AA 01 F5 DB D3 E6 01 CA B6 01 F1 2F D3   .#............/.
01C0=D2 C9 E5 21 F2 BF 46 23 4E 78 CD E1 01 CD B5 01   ...!..F#N.......
01D0=79 0F 0F 0F 0F CD E1 01 CD B5 01 79 CD E1 01 E1   ................
01E0=C9 E6 0F C6 30 E0 C6 80 C9 C5 3A F0 BF FE 00 CA   ....0.....:.....
01F0=0B 02 01 CE 57 E3 E3 E3 E3 E3 E3 0D C2 F5 01 05   ....W...........
0200=C2 F5 01 C6 99 27 FE 00 C2 F2 01 C1 C9 23 7E E6   .....'.......#..
0210=7F E8 C6 80 C9 3E 17 D3 D3 C3 94 00 3E 15 D3 D3   .....>......>...
0220=C3 94 00 F5 DB D1 E6 01 CA 24 02 7E FE 00 CA 37   .........$.....7
0230=02 D3 D0 23 C3 24 02 F1 C9 C5 23 7E E6 0F 0F 0F   ...#.$....#.....
0240=0F 0F 47 23 7E E5 E6 0F B0 47 FE 00 CA 62 02 21   ..G#.....G.....!
0250=65 04 CD AA 01 78 C6 99 27 47 FE 00 CA 62 02 C3   ........'G......
0260=4F 02 E1 C1 23 7E C9 21 70 04 CD 23 02 21 00 80   O...#..!...#.!..
0270=CD 82 01 FE 83 CA 90 02 FE 04 CA 90 02 FE 9E CA   ................
0280=67 02 FE 08 C2 8B 02 2B C3 70 02 77 23 C3 70 02   .......+....#...
0290=77 AF 23 77 C9 21 AC 04 CD 23 02 11 00 00 CD E7   ..#..!...#......
02A0=02 FE FF CA 95 02 FE 0D CA C0 02 47 0E 04 7B 17   ...........G....
02B0=5F 7A 17 57 0D C2 AE 02 7B E6 F0 B0 5F C3 9E 02   _..W........_...
02C0=21 0E 05 CD 23 02 0E 00 CD E7 02 FE FF CA C0 02   !...#...........
02D0=FE 0D CA E2 02 47 79 07 07 07 07 E6 F0 B0 4F C3   .....G........O.
02E0=C8 02 21 F0 BF 71 C9 CD 82 01 E6 7F FE 0D C8 FE   ..!.............
02F0=30 FA FF 02 FE 3A F2 FF 02 CA FF 02 E6 0F C9 3E   0....:.........>
```

```
0300=FF C9 AF 21 F2 BF 77 23 77 C9 DB D5 E6 01 CA 0A   ...!...#........
0310=03 7E FE 00 C8 D3 D4 23 C3 0A 03 DB D7 E6 01 CA   .......#........
0320=1B 03 7E FE 00 C8 D3 D6 23 C3 1B 03 DB D5 E6 02   ........#.......
0330=CA 2C 03 DB D4 C9 DB D7 E6 02 CA 36 03 DB D6 C9   .,..........6....
0340=DB D3 E6 02 CA 56 03 DB D2 2F 77 FE 83 CA 6A 03   .....V.../......
0350=D3 D0 23 C3 40 03 DB EB E6 02 C2 40 03 36 00 3E   ..#.@......@.6.>
0360=94 D3 D3 21 70 05 CD 23 02 C9 E5 21 73 05 CD 23   ...!...#...!...#
0370=02 E1 23 C3 40 03 DB D1 E6 02 CA A0 03 DB D0 D3   ..#.@...........
0380=D0 FE 9B CA 91 03 21 00 80 77 23 CD 70 02 C3 76   ......!...#.....
0390=03 21 00 80 CD 4A 01 CD 1D 01 CD 5D 01 C3 76 03   .!...J.....]....
03A0=DB EB E6 02 CA 76 03 21 70 05 CD 23 02 21 00 A0   .......!...#.!..
03B0=CD 40 03 C3 76 03 0D 8A 52 45 50 45 54 49 54 49   .@......REPETITI
03C0=56 45 20 4D 45 53 53 41 47 45 20 54 45 53 54 2C   VE MESSAGE TEST,
03D0=20 41 53 52 20 45 4D 55 4C 41 54 49 4F 4E 2C 20    ASR EMULATION,
03E0=4F 52 20 53 50 45 43 49 41 4C 20 54 45 53 54 3F   OR SPECIAL TEST?
03F0=0D 8A 28 54 59 50 45 20 52 2F 41 2F 53 29 20 20   ..(TYPE R/A/S)
0400=00 0D 8A 8A 54 59 50 45 20 52 20 54 4F 20 42 45   ....TYPE R TO BE
0410=47 49 4E 20 52 45 47 45 4E 20 54 45 53 54 0D 8A   GIN REGEN TEST..
0420=54 59 50 45 20 4E 20 54 4F 20 42 45 47 49 4E 20   TYPE N TO BEGIN
0430=4E 4F 4E 20 52 45 47 45 4E 20 54 45 53 54 0D 8A   NON REGEN TEST..
0440=54 59 50 45 20 41 20 54 4F 20 41 42 4F 52 54 20   TYPE A TO ABORT
0450=07 00 00 00 00 00 00 00 00 00 57 D5 16 16 00      ...........W....
0460=03 03 03 03 00 03 3C DD 3F 84 41 24 FF C3 16 00   ......<.?.A$....
0470=0D 8A 8A 45 4E 54 45 52 20 54 45 53 54 20 4D 45   ...ENTER TEST ME
0480=53 53 41 47 45 2E 0D 8A 45 4E 44 20 57 49 54 48   SSAGE...END WITH
0490=20 45 54 58 20 4F 52 20 45 4F 54 2E 0D 8A 8A 00    ETX OR EOT.....
04A0=00 00 00 00 00 00 00 00 00 00 00 0D 8A 8A 45      ...............E
04B0=4E 54 45 52 20 54 48 45 20 4E 55 4D 42 45 52 20   NTER THE NUMBER
04C0=4F 46 20 54 49 4D 45 53 20 54 48 45 20 4D 45 53   OF TIMES THE MES
04D0=53 41 47 45 20 49 53 20 54 4F 20 42 45 20 54 52   SAGE IS TO BE TR
04E0=41 4E 53 4D 49 54 54 45 44 2E 0D 8A 45 4E 44 20   ANSMITTED...END
04F0=57 49 54 48 20 22 52 45 54 55 52 4E 22 2E 20 20   WITH "RETURN".
0500=20 00 00 00 00 00 00 00 00 00 00 00 00 0D 8A       ...............
0510=45 4E 54 45 52 20 4E 4F 2E 20 4F 46 20 53 45 43   ENTER NO. OF SEC
0520=4F 4E 44 53 20 4F 46 20 44 45 4C 41 59 20 42 45   ONDS OF DELAY BE
0530=54 57 45 45 4E 20 4D 45 53 53 41 47 45 53 2E 0D   TWEEN MESSAGES..
0540=8A 45 4E 44 20 57 49 54 48 20 22 52 45 54 55 52   .END WITH "RETUR
0550=4E 22 2E 20 20 20 00 00 00 00 00 00 00 00 00 00   N".   ..........
0560=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0570=0D 8A 00 20 45 54 58 00 FF FF FF FF FF FF FF FF   ... ETX.........
```

GLOSSARY

AC	auxiliary carry
AFSATCOM	Air Force Satellite Communications
ASCII	American Standard Code for Information Interchange
ASR	automatic send/receive
BCD	binary coded decimal
CRT	cathode ray tube
CTS	clear to send
DAA	decimal adjust accumulator
DCD	data carrier detect
DIP	dual in-line package
DSR	data set ready
DTR	data terminal relay
EIA	Electronic Industries Association
I/O	input/output
LED	light emitting diode
LSB	least significant bit
MDS	Microcomputer Development System
MSB	most significant bit
PROM	programmable read-only memory
RAM	random access memory
ROM	read-only memory
RTS	request to send
RX	receive
RXC	receive clock
SLDT&E	System Level Development Test and Evaluation
TCC	Test Control Center
TDM	time division multiplex
TI	Texas Instruments
TX	transmit
TXC	transmit clock
USART	universal synchronous-asynchronous receiver-transmitter

FILMED

-8