NASA Technical Memorandum 101375

Design and Implementation of a Microcomputer-Based User Interface Controller for Bursted Data Communications Satellite Ground Terminals

Mary Jo W. Shalkhauser
Lewis Research Center
Cleveland, Ohio

December 1988

NASA

DESIGN AND IMPLEMENTATION OF A MICROCOMPUTER-BASED USER

INTERFACE CONTROLLER FOR BURSTED DATA COMMUNICATIONS

SATELLITE GROUND TERMINALS

Mary Jo W. Shalkhauser
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

SUMMARY

The NASA Lewis Research Center is developing a laboratory-based satellite communications test bed for evaluation of state-of-the-art communications hardware and systems. Most of the digital components of the ground terminals are being constructed in-house at NASA Lewis. One of the ground terminal subsystems, the user interface controller, controls the connection and disconnection of all users to communication network. This paper will describe the role of the user interface controller in the ground terminal and discuss the design and implementation of the microcomputer-based subsystem.

INTRODUCTION

The Space Electronics Division of the NASA Lewis Research Center is currently developing, in hardware and software, an in-house laboratory model of a satellite communications system. This laboratory model, called the Systems Integration, Test, and Evaluation (SITE) Project, is an on-going test bed for the evaluation of state-of-the-art satellite communication components. The SITE system enables NASA to exercise and test microwave components (e.g., matrix switches, receivers, and traveling wave tube amplifiers (TWTA's)), digital components (baseband processors, ground terminals, modulators, demodulators, terrestrial interfaces), and network control algorithms in a realistic satellite system environment (ref. 1).

The laboratory model is currently configured for the Advanced Communications Technology Satellite (ACTS) Ka-band SS-TDMA architecture (fig. 1). Information and expertise gained from the SITE project has provided useful results to the Lewis ACTS Project Office. The SITE project will be used in a similar manner for future NASA satellite communication system developments.

In the SITE communication system, transmitting and receiving users are simulated by a bit-error-rate test set that consists of the data generator and data checker. Simulating a transmitting user, the data generator creates continuous, serial, pseudorandom data at a variety of rates from 1 to 200 Mbps. The data checker simulates a receiving user by performing bit-error-rate checking on the user data after it is transmitted through the satellite and received by the ground terminal. A single transmitting and receiving user is simulated by housing both a data generator and a data checker together in a single chassis. Three of these transmitting and receiving users are connected to each SITE ground terminal. A detailed discussion of the data generator and data checker designs and operation is contained in reference 2.

The experimental operation of the SITE system is provided by the experiment control and monitor (EC&M) computer. The EC&M computer creates different operating environments, including simulated range delay, satellite motion, and rain fade, for the SITE communication system and monitors the status of the system. The EC&M computer also provides intelligent control of the simulated users to create changing user traffic demands and conditions. The control of the users is provided through the EC&M interface microcomputer. A detailed discussion of user simulation and the EC&M interface microcomputer (EIM) is contained in reference 3.

The core of the current SITE model is the digital portion of the ground terminal (see fig. 1). The ground terminal receives continuous serial data from the three simulated users and compresses the data into short bursts ready for transmission with transmit first-in, first-out (FIFO) memories. Each ground terminal is given transmit and receive burst time assignments based on demand and on current network connectivity. The transmitting ground terminal inserts the users' data into assigned time slots of a TDMA frame and bursts the user data to the satellite through a serial minimum shift keying (SMSK) modulator at a high burst rate of nominally 221 Mbps. The receiving ground terminal receives data bursts from the satellite through an SMSK demodulator and routes the data to the proper receiving user. The high rate user data bursts are expanded into a slower rate continuous data stream by the ground terminal receive FIFO memories. Every SITE ground terminal supports both transmitting and a receiving users.

Network control of the communication system is provided by one of the ground terminals in the SITE system called the master control terminal (MCT). The MCT monitors all user traffic within the SITE communication system. Changes in the burst time plan for user traffic assignments within the TDMA frame must be approved by the MCT before implementation. Before a ground terminal can transmit user data, it must request a TDMA time slot from the MCT by sending order wire messages through the communication link to the MCT. Permission or denial of user time slots is sent back to the requesting ground terminal by the MCT through reference burst messages.

Within every SITE ground terminal, the user interface controller links the three simulated users to the ground terminal and controls initiation and termination of all user data transmissions. The ground terminal user interface controller is the topic of this paper.

GLOSSARY

ACK   acknowledge signal

ACTS   Advanced Communications Technology Satellite

CPU   central processing unit

$\overline{\text{CTS}}$ clear to send signal

$\overline{\text{DAV}}$ data available signal

DM   dump memory command

EC&M	experiment control and monitor
EIM	EC&M interface microcomputer
EPROM	erasable programmable read only memory
FIFO	first in, first out
I/O	input/output
MCT	master control terminal
RAM	random access memory
$\overline{RQD}$	request denied signal
$\overline{RTS}$	request to send signal
S68000	send to Motorola 68000 command
SITE	systems integration, test, and evaluation
SM	set memory command
SMSK	serial minimum shift keying
SS-TDMA	satellite-switched time-division multiple-access
TWTA	traveling wave tube amplifier

DESIGN APPROACH

The ground terminal user interface controller has three major functions:

(1) Monitoring the status of the three transmitting and receiving users connected to the ground terminal

(2) Sending user transmission connect and disconnect requests to the ground terminal order wire processor

(3) Receiving messages from the ground terminal order wire processor granting or denying transmission requests and providing the appropriate control signals to the users

The role of the SITE user interface controller in the ground terminal will be described in the following discussion of how a new user enters communication system.

The EC&M computer initiates a user transmission with a start command to the EIM (refer to fig. 2). After receipt of the EC&M message, the EIM creates a start pulse to the data generator (transmitting user). To make a transmission request to the ground terminal, the data generator activates its request-to-send ($\overline{RTS}$) signal. The ground terminal user interface controller

3

continuously monitors the status of each user's $\overline{RTS}$ signal. When the user interface controller sees an active $\overline{RTS}$ signal, it sends a connect request command to the ground terminal order wire processor (a Motorola 68000-based microcomputer).

The order wire processor then creates an order wire message to the MCT requesting the connection of a new user to the system. The MCT checks the current capacity of the network to see if the new transmission can be accommodated. If there is available capacity, the MCT will notify both the source and destination ground terminals of the new connection and the corresponding transmit and receive time division, multiple access time slot assignments through reference burst messages. If there is insufficient available capacity, the MCT will notify the requesting ground terminal that the transmission request has been denied.

All reference burst messages from the MCT to each ground terminal are processed by that ground terminal's order wire processor. User related reference burst information is passed from the order wire processor through a command port to the user interface controller. When appropriate, the user interface controller creates control signals to the requesting user, reflecting the MCT's response to the transmission request.

When the MCT grants a new connection, it first attempts to connect the receiving user at the destination ground terminal to the network via a reference burst message. The destination ground terminal order wire processor then sends a receive connect command to the user interface controller, which activates a data available ($\overline{DAV}$) signal to the appropriate data checker (receiving user). When the simulated user is ready to receive data, it responds with an acknowledge ($\overline{ACK}$) signal to the user interface controller.

To complete the connection, the MCT sends a reference burst message to the source ground terminal. The source ground terminal order wire processor sends a connect command to the user interface controller and a clear-to-send ($\overline{CTS}$) signal directly to the appropriate data generator (transmitting user). Then the data generator begins sending data.

HARDWARE

A block diagram of the user interface controller and all of its hardware connections is shown in figure 3. The ground terminal user interface controller is implemented using a TITN, Inc., microcomputer board model TN 6835. This microcomputer board contains a Motorola 6809 microprocessor central processing unit, 64 KB of on-board dynamic random access memory (RAM), and 128 KB of on-board erasable programmable read-only memory (EPROM). Two RS-232 serial ports and 48 lines of parallel input and output (I/O) are also contained on the microcomputer board (ref. 4). The TN 6835 board is highly user configurable for flexibility, and contains a Multibus I connector for expandibility.

Two Multibus I Intel SBC-519 I/O expansion boards are also part of the user interface controller design. Each SBC-519 board contains three, 24-bit parallel I/O ports. Data direction of each port is configurable for several combinations of input and output lines. The Motorola 6809 microcomputer board and the two I/O expansion boards are all mounted in a powered Multibus chassis.

4

The ground terminal user interface controller connects to each data generator and data checker associated with the ground terminal, the EC&M interface microcomputer, and other subsystems within the ground terminal. A block diagram of the user interface controller and all of its hardware connections is shown in figure 3. All interface connections are provided through the microcomputer on-board parallel I/O ports and the Intel SBC-519 I/O expansion boards. Each parallel I/O port is divided into 8-bit groups and assigned an individual memory address. Input operations are performed in software by loading the value from the assigned memory address (e.g., LDA 0F902H), and output operations are performed by storing a value to the assigned memory address (e.g., STA 0F902H). The configuration of all the I/O ports is listed in table I.

Communication between the user interface controller and the order wire processor is performed through one of the I/O ports on an Intel SBC-519 board. This port is configured for strobed I/O mode (ref. 5) and is connected through a 50-pin ribbon cable to a similar I/O port on the order wire processor. Commands (shown in table II) are passed between the two microcomputers through this port, one 8-bit byte at a time. The 24-bits of I/O of this port are divided into 8 bits for command input, 8 bits for command output, and 8 lines for handshaking and control of strobed data transfer. A photograph of the user interface controller hardware is shown in figure 4.

TABLE I. - USER INTERFACE CONTROLLER INOUT-OUTPUT PORT CONFIGURATION

Board	Port	Group	Address (hex)	Singal	Direction	Connection
TN 6835	1	A	FB00	Rate, type	Input	EC&M interface
		B	FB01	Destination	Input	microcomputer
		C	FB02	Destination	Input	
SBC-519 board 1	1	B	F901	$\overline{\text{RTS}}$	Input	Transmitting users
		C	F902	$\overline{\text{RQD}}$	Output	Transmitting users
	2	A	F904	$\overline{\text{DAV}}$	Output	Receiving users
		B	F905	$\overline{\text{ACK}}$	Input	Receiving users
		C	F906	DAV	Output	Gorund terminal
	3	A	F908	R1RATE	Output	Recieve clock source board
		B	F909	R2RATE	Output	Recieve clock source board
		C	F90A	R3RATE	Output	Recieve clock source board
SBC-519 board 2	1	A	F910	FIFO set point 1	Output	Transmit FIFO memory
		B	F911	FIFO set point 2	Output	Transmit FIFO memory
		C	F912	FIFO set point 3	Output	Transmit FIFO memory
	2	A	F914	FIFO set point 1	Output	Receive FIFO memory
		B	F915	FIFO set point 2	Output	Receive FIFO memory
		C	F916	FIFO set point 3	Output	Receive FIFO memory
	3	A	F918	Command input	Input	Order wire processor
		B	F919	Command output	Output	Order wire processor
		C	F91A	Control	In/Out	Order wire processor

5

TABLE II. – COMMANDS BETWEEN THE USER INTERFACE CONTROLLER AND THE ORDER WIRE PROCESSOR

(a) Commands to user interface controller

Type of command	Command
Connect	1Z XX YY YY └─Burst length └─Source-destination FIFO memory address 0 = asynchronous transmit 1 = asynchronous receive 2 = asynchronous duplex 8 = synchronous transmit 9 = synchronous receive A = synchronous duplex
Connect denied	3X └─User FIFO memory address
Disconnect	4X YY └─User FIFO memory address 0 = transmit burst deletion 1 = receive burst deletion 2 = duplex burst deletion
Echo response	5X YY YY └─Echo data
Echo request	6X YY YY └─Data to be echoed

(b) Commands to order wire processor

Type of command	Command
Connect	1Z XX YY LL LL └─Burst length └─Destination FIFO memory address └─Destination ground terminal └─Source user FIFO memory address 0 = asynchronous transit 1 = asynchronous receive 8 = synchronous transit 9 = synchronous receive
Echo request	5X YY YY └─Data to be echoed
Echo response	6X YY YY └─Echo data

SOFTWARE

The ground terminal user interface controller was written and developed in Motorola 6809 assembly language. The compiled machine code is programmed and stored in EPROM on the 6809 microcomputer board. The software is divided into three modules: (1) a user status polling loop, which monitors the status of all transmitting and receiving users connected to the ground terminal to determine if a user wants to start or stop data transmission, (2) a command processor, which inputs and executes commands from the order wire processor, and (3) a monitor program, which allows an engineer to interact with the user interface controller software for trouble-shooting purposes.

When the microcomputer chassis is powered-up, a power-on-reset circuit on the microcomputer board creates a restart interrupt to begin program execution. A reset push-button switch on the front of the Multibus chassis also creates the restart interrupt. The program begins with an initialization sequence followed by the user polling routine. The program remains in the user polling

routine loop until an interrupt occurs. Two actions can cause an interrupt: a command sent to the user interface controller from the order wire processor and a keyboard input to the monitor program. The order wire processor commands are assigned to the highest priority interrupt, so that they will be processed quickly. Only the nonmaskable restart interrupt has priority over the order wire processor commands.

User Status Polling Module

The polling routine (fig. 5) is a program loop which checks the transmission status of each user starting with the three transmitting users in order from user 1 to user 3, followed by the receiving users, and then back to the transmitting users. Separate transmit and receive status flags are maintained for each user and are updated as the status changes.

Transmitting users. - The status polling routine first checks the transmit status flag to see if the user was transmitting data at the last status check. If not, the program tests the $\overline{RTS}$ signal to see if the user wishes to begin a new data transmission. If the $\overline{RTS}$ is active, the program jumps to a subroutine to process the user's transmission request. If the $\overline{RTS}$ is not active, then the user interface controller takes no action.

The order and timing of the events that occur when connecting a new user data transmission to the communication network is shown in figure 6. Once a user requests a new transmission, the user interface controller begins to gather information about the request from the user. The program reads a parallel port on the microcomputer board to acquire the data rate, data type, and destination user number from the EC&M interface microcomputer (EIM). The data rate is represented by a 4-bit code. Table III shows the 16 available data rates for each simulated user and the data rate code assigned to each rate.

TABLE III. - USER DATA RATES

Data rate code (hex)	Rate[a], MHz		Description
	Asynchronous	Synchronous	
0	1.25	1.3824	General purpose
1	1.544	--------	T1 telecommunications rate
2	--------	--------	(b)
3	5.0	5.5296	General purpose
4	6.132	--------	T2 telecommunications rate
5	12.5	13.824	General purpose
6	16.1079	--------	Digital video
7	--------	--------	(b)
8	25.0	27.648	General purpose
9	27.648	27.648	General purpose
A	42.95	--------	Digital video
B	44.736	--------	T3 telecommunications rate
C	50.0	55.296	General purpose
D	100.0	110.592	General purpose
E	--------	--------	(b)
F	200.0	221.184	General purpose

[a]Changes from asynchronous to synchronous require a manual oscillator change.
[b]Not applicable.

TABLE IV. - USER DATA TYPES

Code	Definition
0	Asynchronous simplex
1	Asynchronous duplex
8	Synchronous simplex
9	Synchronous duplex

The data type can be either simplex or duplex and is represented by a 4-bit code. The definitions of these codes are shown in table IV. A synchronous simplex or duplex data type is used when the user's continuous data clock rate is an integer multiple of the ground terminal's burst clock of 221.184 MHz. When the user and ground terminal clocks are synchronous, the user's burst length will be the same every frame. An asynchronous data type is used when the user's data rate is not an integer multiple of ground terminal's burst clock, and the number of user data bits in the burst will vary from frame-to-frame (ref 6). An asynchronous data transmission requires a 64-bit word in every burst to indicate the length of the data portion of the burst.

Although the user interface controller can accommodate duplex transmission requests, the remainder of the ground terminal subsystems cannot, as currently configured. An artificial duplex transmission can, however, presently be created with two simultaneous simplex transmissions between two users. Duplex transmissions will not be discussed further in this paper.

The data destination is a 16-bit code word in which each position where a bit is set represents a desired destination user. As shown in figure 7, each bit represents a different destination ground terminal and user. Presently, up to 16 users distributed across 5 terminals can be addressed. This format allows for single or multiple receiving users (broadcast mode), although the current implementation of the ground terminal allows for only single user destinations.

From the data rate code the user interface controller calculates the required burst length for the new user transmission and determines if there is enough local modulator capacity to accommodate the new user. Local modulator capacity is determined by subtracting all the active transmitting users' data rates from the total available capacity. The combined data rates of the users connected to a single ground terminal cannot exceed 200 Mbps. If the available local modulator capacity cannot accommodate the new transmission, the user interface controller will respond with a local request denied ($\overline{RQD}$) to the requesting user and then return to the status polling routine.

If the local modulator capacity can accommodate the new transmission, the user interface controller derives the destination ground terminal address and user number from the destination code. Then a set point for control of the transmit FIFO is calculated from the data rate and output to the ground terminal. Next, the connection request command (see table II) is formed for subsequent transmission to the order wire processor. One byte of the connection request command is transferred over the strobed I/O port to the order wire processor. The microcomputer waits for the byte to be accepted by the order wire processor before the next byte is transferred. When all bytes of the command

8

have been transferred and accepted, the program returns to the user status polling routine. A flow chart of the new transmission subroutine is contained in figure 8.

The order wire processor receives the connection request command from the user interface controller and transmits a corresponding order wire message to the MCT, through the communication link, requesting a frame allocation to accommodate the new user transmission request. If the MCT determines that there is sufficient system capacity to accommodate the request, it sends a reference burst message to the destination ground terminal to connect the receiving user to the system before it connects the transmitting user. At the destination terminal, the order wire processor transmits a receive connect command (see table II) to the destination ground terminal user interface controller through the command port. The user interface controller is interrupted by the command from the order wire processor, causing the program to jump from the status polling routine to a command input routine. The user interface controller then inputs and decodes the command. A flow chart of the receive transmission connect subroutine is shown in figure 9. First, the program calculates the data rate and the FIFO set point from the burst length and outputs them through parallel I/O ports to other ground terminal subsystems. Then, the user interface controller gives a $\overline{DAV}$ signal to the appropriate receiving user, waits for the user to acknowledge ($\overline{ACK}$ signal), and returns to the status polling routine.

To complete the new transmission connection, the MCT approves new data transmission by sending a reference burst message to the source ground terminal. The order wire processor in the source ground terminal then sends a transmit connect command to the user interface controller and a $\overline{CTS}$ signal to the appropriate transmitting user. The user interface controller exits the status polling routine to input and process the connect command. The status flags are updated, and the program returns to the status polling routine.

When a transmit status flag shows that the user was transmitting as of the last status check, the user interface controller reads the $\overline{RTS}$ signal from the user to see if the user is still transmitting. If the $\overline{RTS}$ signal is still active, the user is still transmitting, and no action is required. If the $\overline{RTS}$ is inactive, then the user has stopped transmitting, and the user interface controller must disconnect that user from the communication system. At this point, the program jumps to a subroutine to disconnect the user.

A user disconnection is processed by first updating the status flags and then clearing the burst length variable. A disconnect command is identical to a connection request command with a burst length equal to zero. The command is formed and transmitted to the order wire processor, and the program then returns to the status polling routine.

The order wire processor notifies MCT of the disconnect request through order wire communications. The MCT then notifies the destination ground terminal of the user disconnect via a reference burst. The destination terminal's order wire processor transmits a disconnect command (see table II) to the user interface controller. The user interface controller is interrupted by the command from the order wire processor, causing the program to jump from the status polling routine to a command input routine where the disconnect command is input and decoded. If a correct disconnect command is received from the order wire processor, then the user interface controller removes the $\overline{DAV}$ signal to the receiving user. The receiving user removes the $\overline{ACK}$ signal in response.

Receiving Users. – After all three transmitting users requests are serviced, the status of the three receiving users is checked. The user interface controller first tests the receive status flag to see if the user was receiving data at the last status check. If the user was not previously receiving, no action is taken, and the status of the next receiving user is tested. If the receiving user was previously receiving, the $\overline{ACK}$ signal is tested. If the $\overline{ACK}$ signal is still active, the user is still receiving data, and no action is taken. If the $\overline{ACK}$ signal is inactive, the user has stopped receiving data, and the transmission must be removed from the communication network. The destination terminal user interface controller forms and sends a receive disconnect command to its order wire processor. As with a transmit disconnect, the receive disconnect is a connect request command with a zero burst length.

After obtaining a receiving user disconnect command from the user interface controller, the destination terminal order wire processor notifies MCT of the disconnect with an order wire message. The MCT sends a reference burst message to the source ground terminal notifying it of the receiving user's disconnection. In response, the source ground terminal order wire processor gives a disconnect command to the source user interface controller for the appropriate transmitting user.

As before, the command from the order wire processor causes an interrupt in the user interface controller. The user interface controller jumps from the status polling routine to input and decode the command. If a correct disconnect command is received, the user interface controller updates the status flags. Then the ground terminal removes the $\overline{CTS}$ signal to the transmitting user. The user responds by stopping transmission and removing the $\overline{RTS}$ signal.

Command Processor Module

The command processor module inputs and executes commands from the order wire processor. The first byte of a command sent to the user interface controller from the order wire processor causes an interrupt. The program will pause current execution to service the interrupt through the command processor routine (see fig. 10).

Commands received from the order wire processor are input one byte at a time. From the first byte, the total number of bytes in the command can be determined. A look-up table is used to derive the command length from the first byte. All bytes in the command are then input and stored in a command buffer.

When the whole command is received, it is processed to determine if it is a valid command (see table II). If it is valid, the command is executed. If it not valid, an error message is printed on a monitor terminal, and the command is ignored. The execution of the commands from the order wire processor is described in the sections "User Status Polling Module" and "Monitor Module."

Monitor Module

The monitor routine and keyboard enables an engineer to interact with the user interface controller software during execution. An ASCII terminal is connected to the microcomputer through one of the RS-232 serial communication

10

ports of the microcomputer board. Any character entered on the monitor keyboard causes an interrupt, and the program jumps out of the status polling routine and into the monitor routine. However, because the command processor module has a higher priority interrupt than the monitor module, any command processor module will finish before the monitor interrupt is serviced.

Through the monitor terminal, the engineer can read the contents of the programmable read only memory (PROM) and the random access memory (RAM) with a display memory (DM) command, read and change the contents of RAM with a set memory (SM) command, execute user interface controller software with a GO command, and send commands to the order wire processor with a SEND (S68000) command. The monitor commands and their formats are shown in table V. A flow chart of the keyboard command software is shown in figure 11.

The command interface between the user interface controller and the order wire processor can be functionally tested through the monitor program. An ECHO command containing two bytes of data can be sent manually to the order wire processor. The exact two bytes of data are sent back to the user interface controller in an echo response command from the order wire processor. The data received in the echo response command is displayed on the terminal for comparison with the transmitted data in the first echo command.

The monitor also displays error messages for trouble-shooting purposes. Error messages are displayed because of improper commands typed on the monitor terminal, invalid commands received from the order wire processor, invalid changes in control signals from users, and invalid user information from a transmitting user.

STATUS

The user interface controller subsystem is complete and has been successfully integrated and tested with the SITE ground terminal. Three units have been constructed and integrated: two in SITE ground terminals and one in a high-burst-rate link evaluation terminal for the ACTS project. The user interface controller software allows for future expansion, including duplex and broadcast mode data transmissions.

TABLE V.- MONITOR COMMANDS

Command	Definition
S68000 XX XX XX XX	Send command XX XX XX XX to order wire processor
SM XXXX	Display and update data at memory location XXXX
DM XXXX YYYY	Dump memory contents from address XXXX to address YYYY
GO XXXX	Execute the program code starting at address XXXX

CONCLUDING REMARKS

The System Integration, Test, and Evaluation (SITE) Project of the NASA Lewis Research Center is currently constructing a laboratory model of a Ka-band satellite-switched time-division multiple-access (SS-TDMA) satellite system. The model's ground terminal contains a subsystem called the user interface controller. Realistic transmitting and receiving users of the communication system are simulated with a bit-error-rate test set called the data generator and data checker. The ground terminal user interface controller regulates the simulated users' requests for data transmission connections and disconnections.

The user interface controller monitors the status of the three transmitting and three receiving users connected to the ground terminal. If a user wishes to begin or end a data transmission, a status change is detected. Through the ground terminal order wire processor, the user interface controller must obtain permission for all changes in user traffic from one special ground terminal in the SITE system called the master control terminal (MCT). The user interface controller requests user traffic changes through a command port to the order wire processor. The order wire processor communicates to the MCT with order wire (to MCT) and reference burst (from MCT) messages through the communication link. The MCT's response to the order wire processor is transmitted back to the user interface controller through the command port. The user interface controller then creates the appropriate control signals to the transmitting and receiving users.

The user interface controller is implemented using a Motorola 6809-based microcomputer. A multibus connector on the microcomputer board allows easy input and output expansion. The software was written and developed in 6809 assembly language, and was compiled and stored in on-board erasable programmable read-only memory. The software was written in three modules: (1) the user status polling module, which monitors user output signals for requests for connections or disconnections, (2) the command processor module, which receives and performs commands from the order wire processor, and (3) the monitor module, which enables an engineer to interact with the user interface controller software, while it is executing, for trouble-shooting.

Presently, three user interface controller subsystems have been constructed and successfully integrated in SS-TDMA ground terminals.

REFERENCES

1. Bagwell, J.W.: A System for the Simulation and Evaluation of Satellite Communication Networks. 10th Communications Satellite Systems Conference, AIAA, 1984, pp. 172-180.
2. Windmiller, M.J.: Unique Bit-Error-Rate Measurement System for Satellite Communication Systems. NASA TP-2699, 1987.
3. Shalkhauser, M.J.: Satellite Ground-Terminal User Simulation. NASA TM-100234, 1988.
4. TN 6835 User's Guide (Technical). TITN Inc., July 1983.
5. Intel Microsystem Components Handbook, Vol II., Intel Corp., Santa Clara, CA, 1984, p. 6-358.
6. Feher, K.: Digital Communications, Satellite/Earth Station Engineering. Prentice-hall, 1983, pp. 347-349.
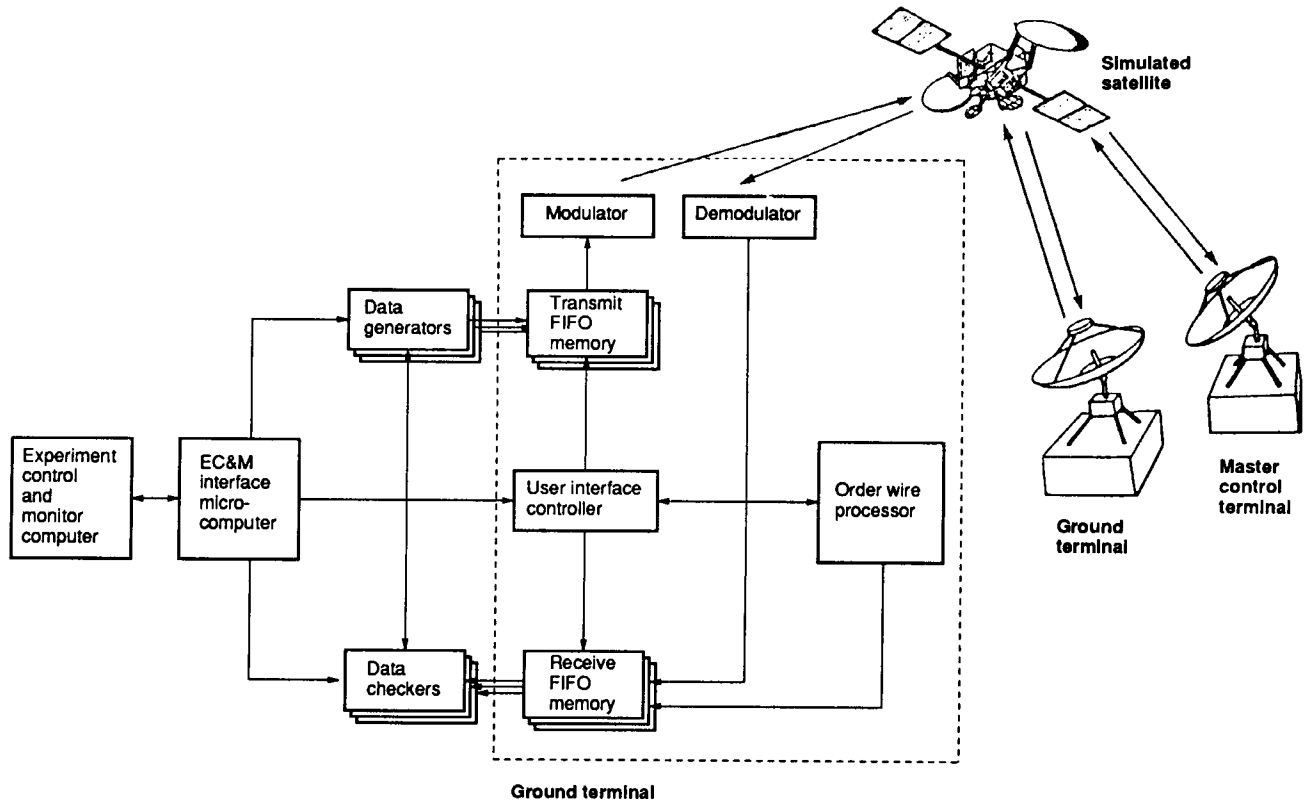
Figure 1. - SITE Project communication system configuration.
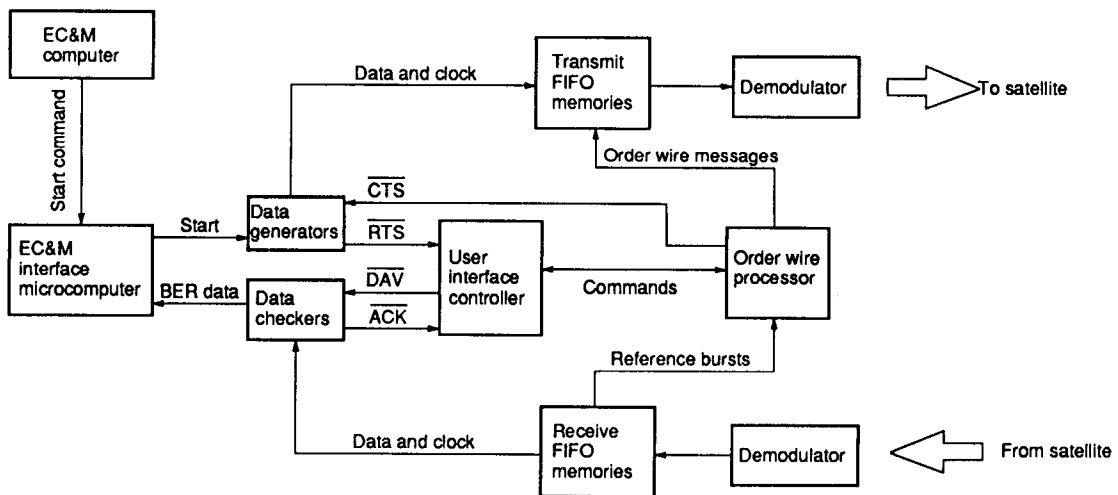
Figure 2. - SITE ground terminal operational block diagram (shows only one data generator and checker simulated user).
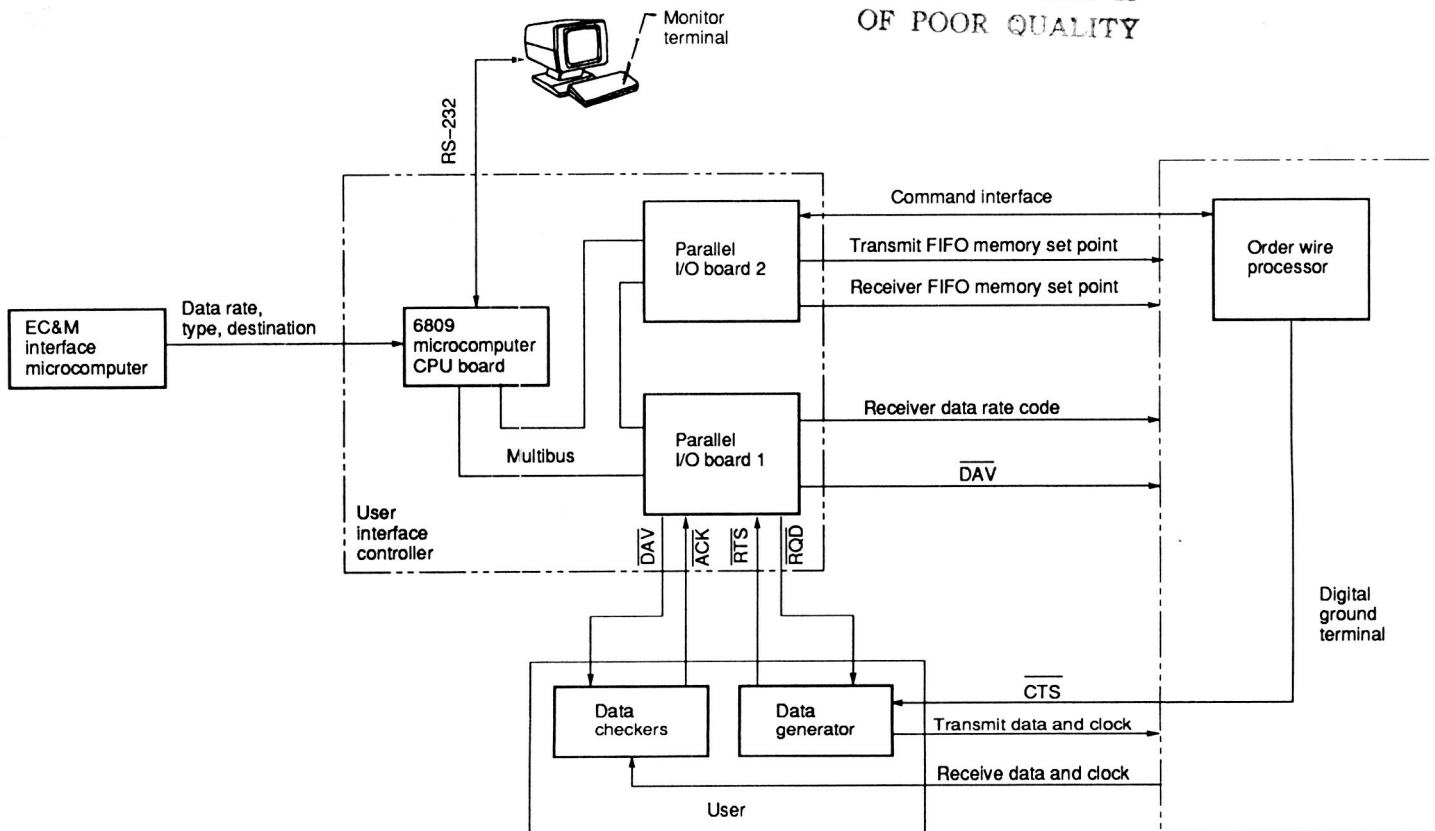
Figure 3. - User interface controller block diagram.
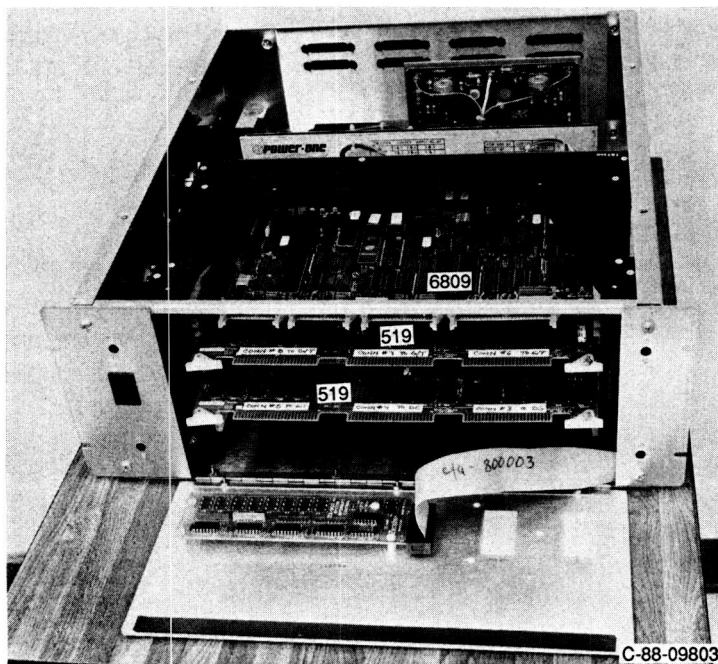
Figure 4. - User interface controller.
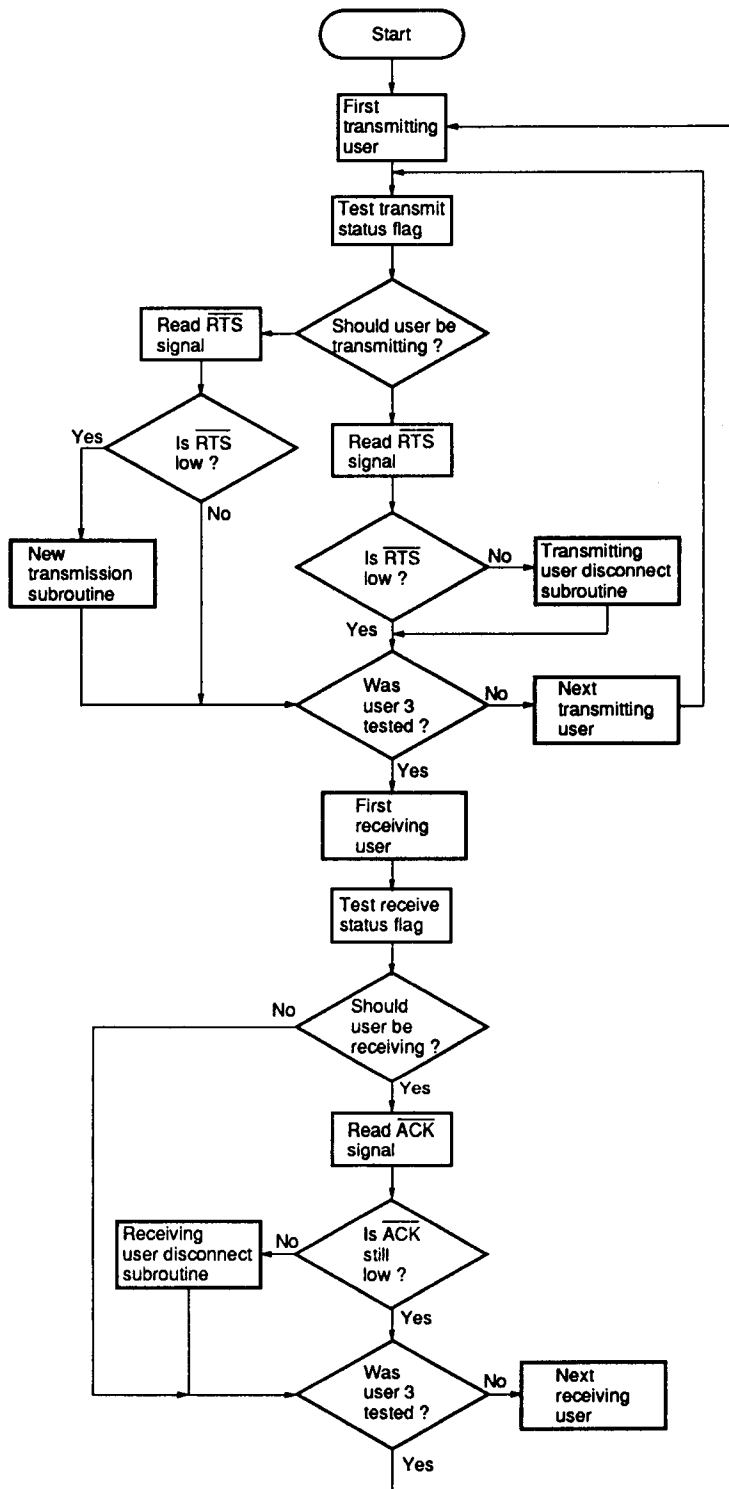
Figure 5. - User polling routine flow chart.
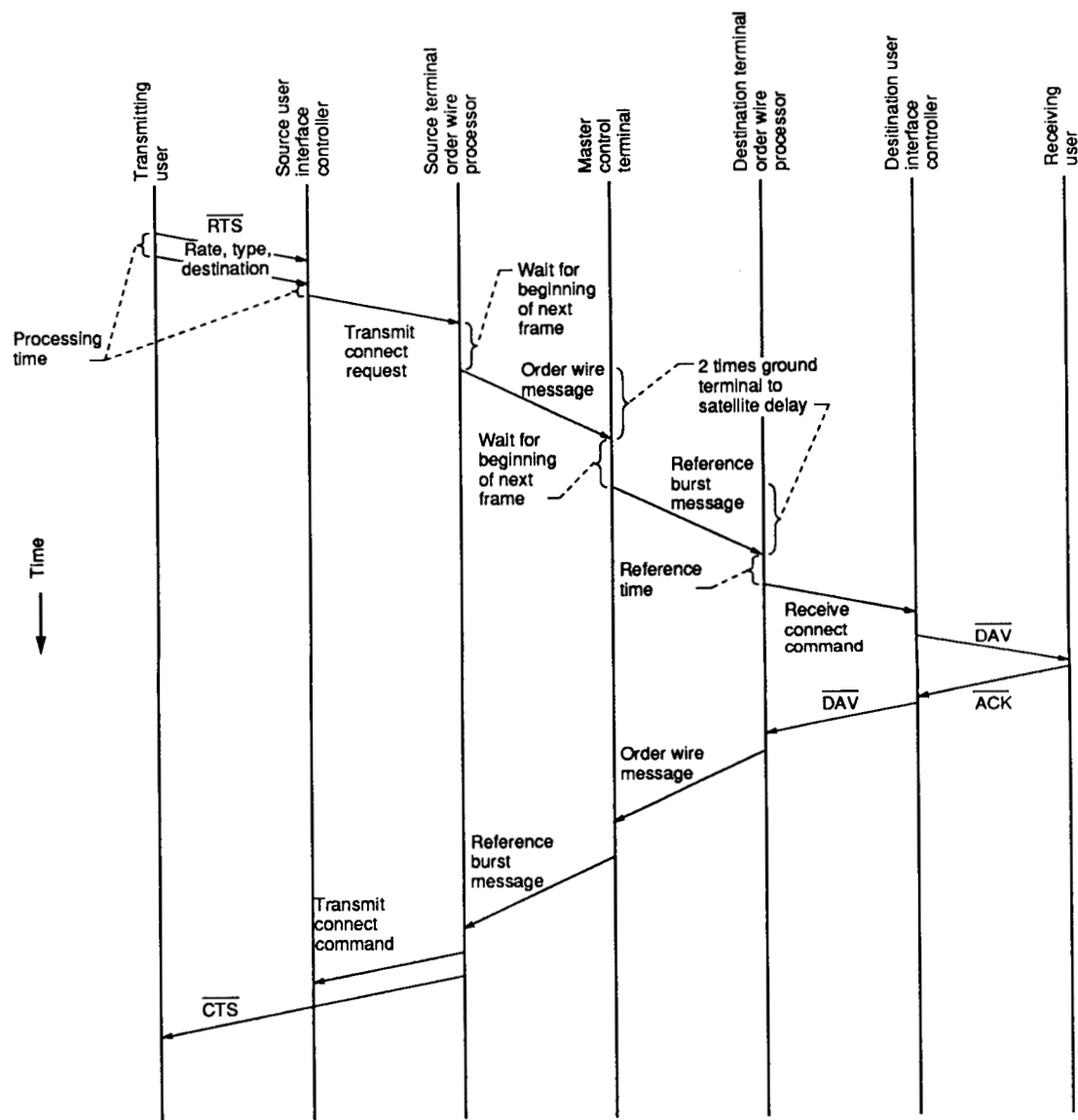
Figure 6. - Timing of events for a user request for a new data transmission.

Ground terminal

| 5 | | 4 | | | 3 | | | 2 | | | 1 | | | 0 | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

← Bit number

User 1
User 2
User 3
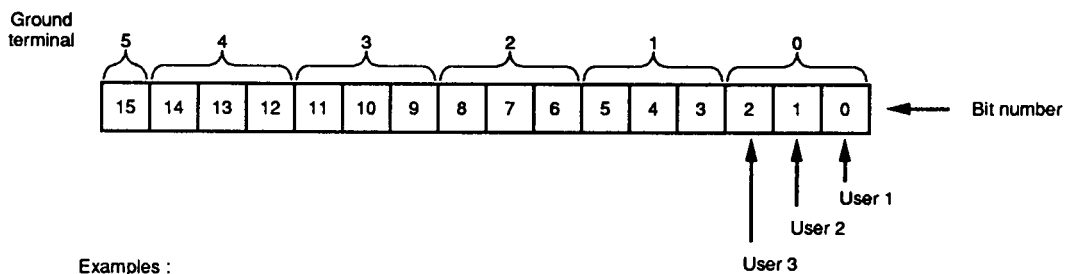
Examples :

0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

Destination user = user 3 on ground terminal 3

0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0

Destination user = user 3, ground terminal 2 and
user 2, ground terminal 1

Figure 7. - Destination user code word definition.

```
                    ┌─────────┐
                    │  Start  │
                    └────┬────┘
                         │
              ┌──────────┴──────────┐
              │ Read data rate      │
              │ and data type       │
              └──────────┬──────────┘
                         │
              ┌──────────┴──────────┐
              │ Read destination    │
              └──────────┬──────────┘
                         │
              ┌──────────┴──────────┐
              │ Calculate burst     │
              │ length from         │
              │ rate                │
              └──────────┬──────────┘
                         │
              ┌──────────┴──────────┐
              │ Read current        │
              │ modulator           │
              │ capacity            │
              └──────────┬──────────┘
                         │
              ┌──────────┴──────────┐
              │ Subtract new rate   │
              │ from modulator      │
              │ capacity            │
              └──────────┬──────────┘
                         │
                    ╱────┴────╲          ┌──────────────────┐
                   ╱ Is new    ╲   Yes   │ Add new rate back│
                  ╱  modulator  ╲────────▶│ into modular     │
                  ╲  capacity   ╱         │ capacity         │
                   ╲  < 0 ?    ╱          └────────┬─────────┘
                    ╲────┬────╱                    │
                         │ No               ┌──────┴──────┐
              ┌──────────┴──────────┐       │ Send local  │
              │ Determine           │       │ RQD signal  │
              │ destination         │       └──────┬──────┘
              │ user                │              │
              └──────────┬──────────┘        ┌─────┴──────┐
                         │                   │ Return from│
              ┌──────────┴──────────┐        │ subroutine │
              │ Calculate transmit  │        └────────────┘
              │ FIFO memory         │
              │ set point           │
              └──────────┬──────────┘
                         │
              ┌──────────┴──────────┐
              │ Output              │
              │ FIFO memory         │
              │ set point           │
              └──────────┬──────────┘
                         │
              ┌──────────┴──────────┐
              │ Form connect        │
              │ request command     │
              └──────────┬──────────┘
                         │
              ┌──────────┴──────────┐
              │ Send first byte     │
              │ of command          │
              └──────────┬──────────┘
                         │
            ┌────────────┤
            │       ╱────┴────╲
            │  No  ╱   Byte    ╲
            └─────◀   accepted ? ╲
                   ╲            ╱
                    ╲────┬────╱
                         │ Yes
            ┌───────────▶│
            │       ╱────┴────╲          Yes
            │      ╱  Command  ╲────────────┐
            │     ╱  all sent ? ╲           │
            │     ╲            ╱            │
            │      ╲────┬────╱      ┌───────┴──────┐
            │           │ No        │ Return from  │
            │  ┌────────┴───────┐   │ subroutine   │
            │  │ Send next      │   └──────────────┘
            │  │ command byte   │
            │  └────────┬───────┘
            │           │◀──────────────┐
            │      ╱────┴────╲           │
            │ Yes ╱           ╲   No     │
            └────◀ Byte accepted ?──────┘
                  ╲           ╱
                   ╲────┬────╱
```

Figure 8. - New transmission subroutine flow chart.

18

Figure 9 flowchart

Start

Determine destination user from command

Is user already receiving ? — Yes → Print error message on monitor terminal → Return from subroutine

(No)

Update receive status flag

Calculate data rate from burst length

Calculate FIFO memory set point from data rate

Output FIFO memory set point

Output $\overline{DAV}$ signal to destination user

Wait for $\overline{ACK}$ signal

Return from subroutine

Figure 9. - Receive–side transmision connect subroutine flow chart.

Figure 10 flowchart

Start

Input first byte

Point to beginning of command buffer

Determine command length from 4 most significant bits of 1st byte

Store in the command buffer

Is complete command input ? — Yes → Decode command → Execute command → Return from interrupt

No

Increment pointer to command buffer

Read command port status word

Is another command byte available ? — No (loops back)
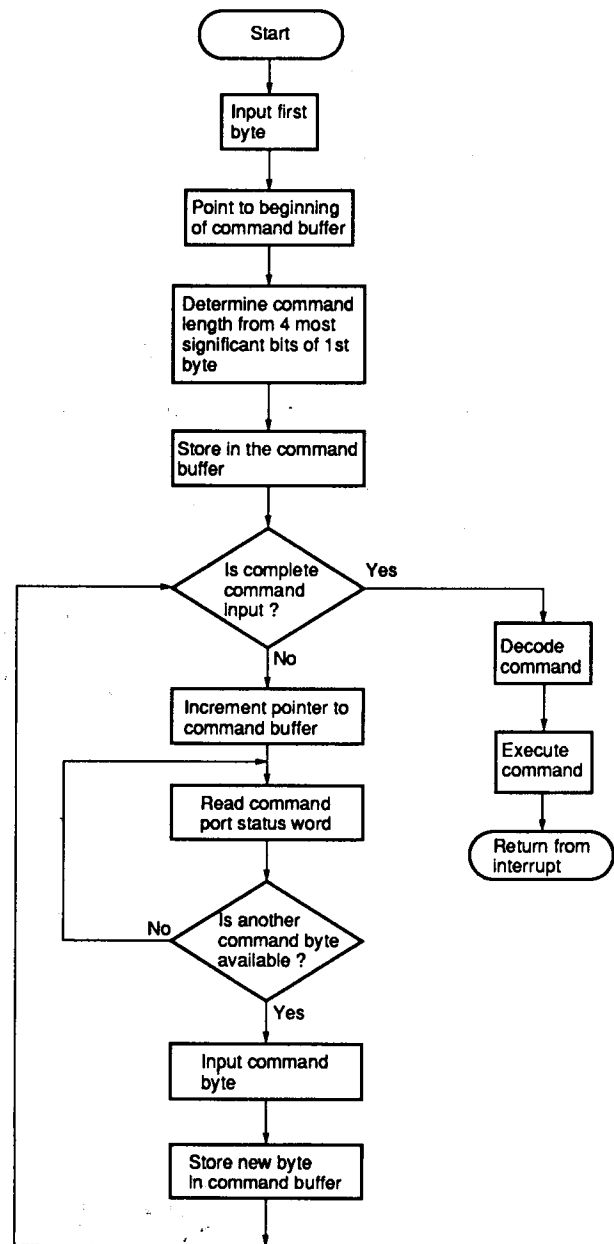
Yes

Input command byte

Store new byte in command buffer

Figure 10. - Command processor subroutine flow chart.

19
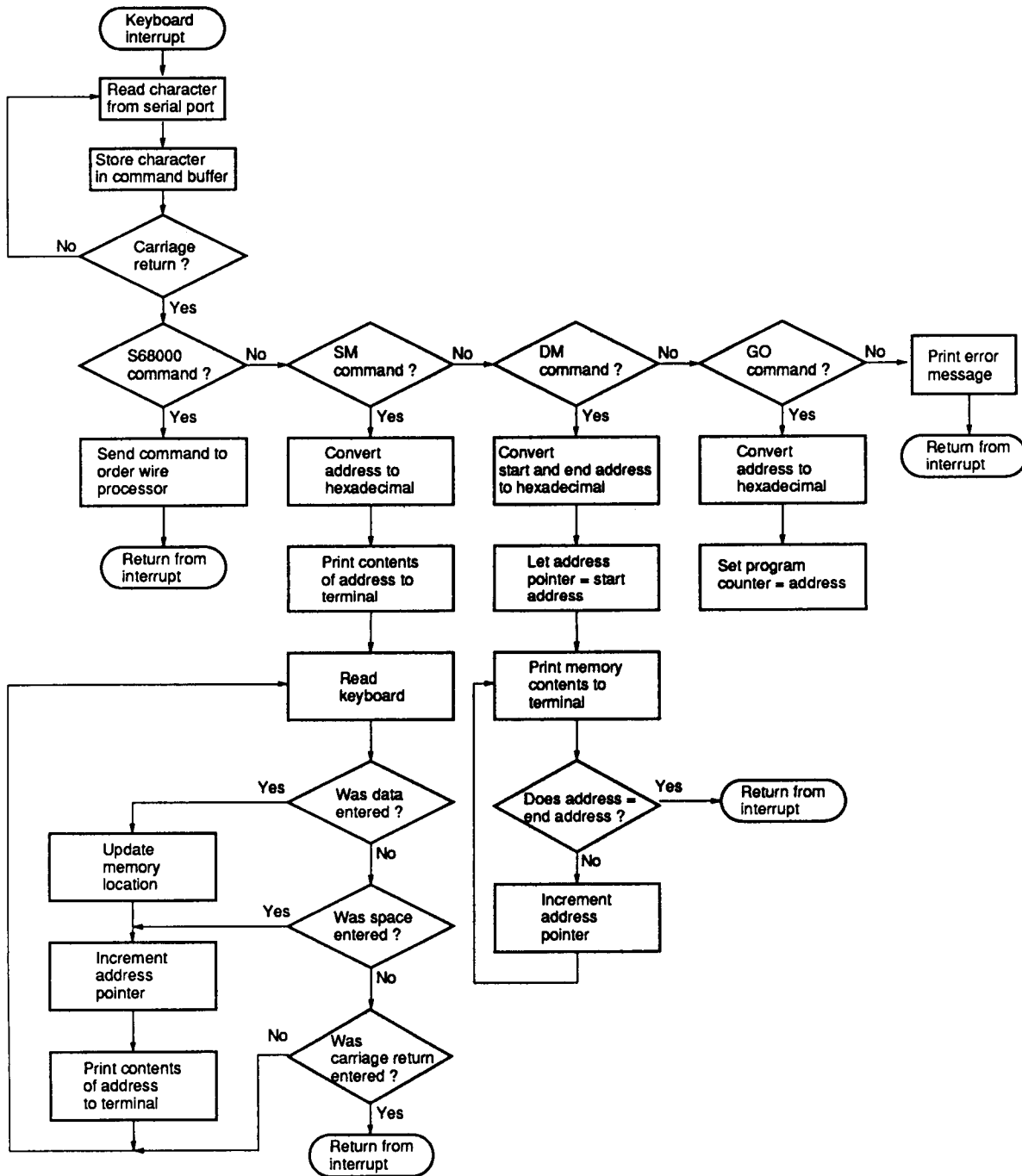
Figure 11. - Monitor module keyboard interupt flow chart.

20

Report Documentation Page

1. Report No. NASA TM-101375	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
Design and Implementation of a Microcomputer-Based User Interface Controller for Bursted Data Communications Satellite Ground Terminals	December 1988
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Mary Jo W. Shalkhauser	E-4428
	10. Work Unit No. 650-60-23

9. Performing Organization Name and Address	11. Contract or Grant No.
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	
	13. Type of Report and Period Covered

12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The NASA Lewis Research Center is developing a laboratory-based satellite communications test bed for evaluation of state-of-the-art communications hardware and systems. Most of the digital components of the ground terminals are being constructed in-house at NASA Lewis. One of the ground terminal subsystems, the user interface controller, controls the connection and disconnection of all users to communication network. This paper will describe the role of the user interface controller in the ground terminal and discuss the design and implementation of the microcomputer-based subsystem.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
User interface Satellite Ground terminal Microcomputer	Unclassified – Unlimited Subject Category 17

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	22	A03