MITRE-Bedford

MTR-4733

JSC 16321

The Crew Activity Planning System Bus Interface Unit

M. A. Allen

DECEMBER 1979

MITRE

MITRE Technical Report
MTR-4733

The Crew Activity Planning System Bus Interface Unit

M. A. Allen

DECEMBER 1979

CONTRACT SPONSOR NASA/JSC
CONTRACT NO. F19628-80-C-0001 T5295F
PROJECT NO. 847C
DEPT. D74

THE MITRE CORPORATION
HOUSTON, TEXAS

Department Approval: Edwin S Hemdon

MITRE Project Approval: Edwin S Hemdon

ABSTRACT

This document details the hardware and software designs used to implement a high speed parallel communications interface to the MITRE 307.2 kilobit/second serial bus communications system from the Harris Universal Block Controller and the Sanders Graphic 7 display terminal. The primary topic is the development of the Bus Interface Unit (BIU), however, discussion of the Harris Slash 8 Universal Block Controller and the Sanders Graphic 7 Parallel Interface Board are used to indicate the interfacing requirements.

TABLE OF CONTENTS

TABLE OF CONTENTS

(Continued)

TABLE OF CONTENTS

(Continued)

TABLE OF CONTENTS

(Continued)

TABLE OF CONTENTS

(Continued)

TABLE OF CONTENTS

(Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

(Continued)

LIST OF ILLUSTRATIONS

(Concluded)

LIST OF TABLES

CREW ACTIVITY PLANNING SYSTEM (CAPS)

BUS INTERFACE UNIT

COMMUNICATIONS HARDWARE AND SOFTWARE

SECTION I

INTRODUCTION

1.0 BACKGROUND

To meet the planning requirements of the Space Shuttle operational schedule, NASA's Johnson Space Center has defined the automated Crew Activity Planning System (CAPS). With a projected flight schedule of up to 52+ flights per year, it was determined that the existing manual planning system used during the Apollo and Skylab projects could not meet the increased demand for crew scheduling [1]. Development of an interactive computer system was begun in 1977 by the McDonnell Douglas Corporation using a Harris Slash 7 computer and several Sanders Associates Graphic 7 display systems.

The original elements of the graphic display system were connected using a standard RS-232C serial interface operating at 9600 bits/second. Separate communication lines were used between each display and the Harris host computer. Initially, all of the terminals were collocated, however, one or more of the Graphic 7 displays will be located in the mission control area approximately one half mile from the host computer when the entire system is made operational. To enable the specified number of terminals to be supported by the Harris host computer, the Slash 7 CPU was replaced by a Slash 8 CPU.

During the development of application software for the CAPS package, it was determined by NASA that the generation of graphic displays was not meeting the response time requirements of the user. A MITRE study [2] of the communications interface, initiated in the fall of 1978, revealed that the Sanders serial interface requires that data, transmitted to a serial port, be padded with additional bits to

insure that all characters are in the ASCII format. This forced an unexpectedly high amount of overhead on the channel and slowed the interface operation. In addition, several key displays required a higher number of bits to be transmitted than had been originally estimated.

One of the study's recommendations for solving the speed deficiency was that a parallel interface be installed between the Harris Slash 8 and the Sanders Graphic 7 displays. This proposal, however, necessitated the installation of some form of high speed communications link to allow for remote display of data in the mission control area. Parallel communication with any computer system is limited to local (within 200 feet) devices by timing factors related to signal propagation.

1.1 Goal

NASA selected the parallel option of the study's recommendations, thus imposing the requirement for a high speed communications medium to transmit data to displays approximately one half mile from the host. The resulting development, documented here, took the approach of inserting a transparent communications medium, the MITRE Coaxial Cable Communications System, between the elements of an existing operational parallel interface for the Harris family of computers and the Sanders Graphic 7 display.

1.2 Scope

The design of a high speed bus communications system to meet the remote communications needs of the CAPS project was initiated in the spring of 1979. This document describes both the hardware and software of the Bus Interface Unit (BIU) used to connect both the Harris Slash 8 and the Sanders Graphic 7 devices to the coaxial cable. In addition, background information on the host computer interface and the Sanders Graphic 7 parallel interface is provided.

Part I of this document (Sections II and III) describes the architecture of the combined CAPS and bus communications system. In Section II, a basic discussion of the functions of the CAPS architecture and communications protocol is presented for background. Details of the individual elements are not discussed; however, the foundation for an understanding of the CAPS-BIU interface is established. Section III

provides a discussion of the basic functions of the cable communications system along with details concerning the interconnection of the elements. The communication protocol used on the cable system is discussed in detail.

Part II (Sections IV and V) describes the hardware used to implement the BIU, Harris and the Graphic 7 interfaces. Section IV is concerned with the details of the Harris Universal Block Controller (UBC) and the Sanders Parallel interface board, both of which are used as parallel ports in the Direct Memory Access (DMA) mode and Teletype (TTY) mode. Section V describes the hardware used in the construction of the MITRE-designed Bus Interface Unit (BIU).

Part III (Sections VI through IX) is a discussion of the software developed for the Harris interface and the two BIU interfaces. The first section in this part of the document is a discussion of the special handler software developed for the Harris interface and a simple introduction to the Graphic 7 handler. The new Harris handler software was developed by NASA by extensive modification and expansion of the existing Harris TTY handler. The DMA portions of the handler were modeled after an existing interface handler developed by MIT Lincoln Labratory. Section VII is a detailed discussion of the BIU software which is common to both the Harris BIU and the Graphic 7 BIU. In Section VIII, a description of the special routines unique to the Harris BIU is presented. In a like manner, Section IX is the detailed description of the routines unique to the Graphic 7 BIU.

Part IV (Sections X and XI) provides the reader with a guide to troubleshooting the CAPS bus communications system and the Bus Interface Unit. Easy to follow steps, which may be used to isolate component failures in the cable plant, are provided in Section X. No information on the repair of these components is provided, rather, test and replacement guidelines are detailed. Section XI provides a limited guide to the isolation of faulty modules within the BIU. By using these guidelines the reader may employ the details in Section V to perform digital checkout of the module's integrated circuits or use chip substitution to return the unit to an operational state.

PART I

BASIC SYSTEM ARCHITECTURE

SECTION II

CREW ACTIVITY PLANNING SYSTEM

2.0 INTRODUCTION

The automated Crew Activity Planning System is designed to allow for the interactive display and modification of flight crew activity schedules for future space flights or the display and modification of the schedules during a mission in progress. The planner uses a Sanders Associates Graphic 7 display station to select various activity menus for display and modification. Each planner operates from a common system data base and a unique user-owned data base. These data bases contain information about flight paths, activity timing, and other background data needed to plan each space shuttle crewperson's daily activity. The data bases are maintained on the Harris S-500 computer system and are managed by interactive application software also running on the Harris host. Communications between the Harris S-500 and each Sanders Graphic 7 is accomplished over a dedicated communications channel using both parallel DMA and parallel TTY techniques. A functional block diagram of the associated interfaces is presented in Figure 2.1-1.

2.1 Harris S-500 Host Computer

The Harris S-500 host computer system is composed of the Harris Slash 8 central processing unit (CPU), control console, two magnetic tape drive units and controllers, two high speed disk pack units and controllers, and various peripheral displays and controllers or interface units. All of the computer hardware is centrally located in Room 274 on the second floor of Building 4 at the NASA Johnson Space Center. Standard Harris procedures are used to interconnect the various hardware units composing the host computer system.

2.1.1 Harris Functions

The Harris S-500 computer system is the host for CAPS operations. The main functions performed by the Harris include control and execution of the CAPS application software, file management of the user and application data bases, data conversion for presentation to the Sanders

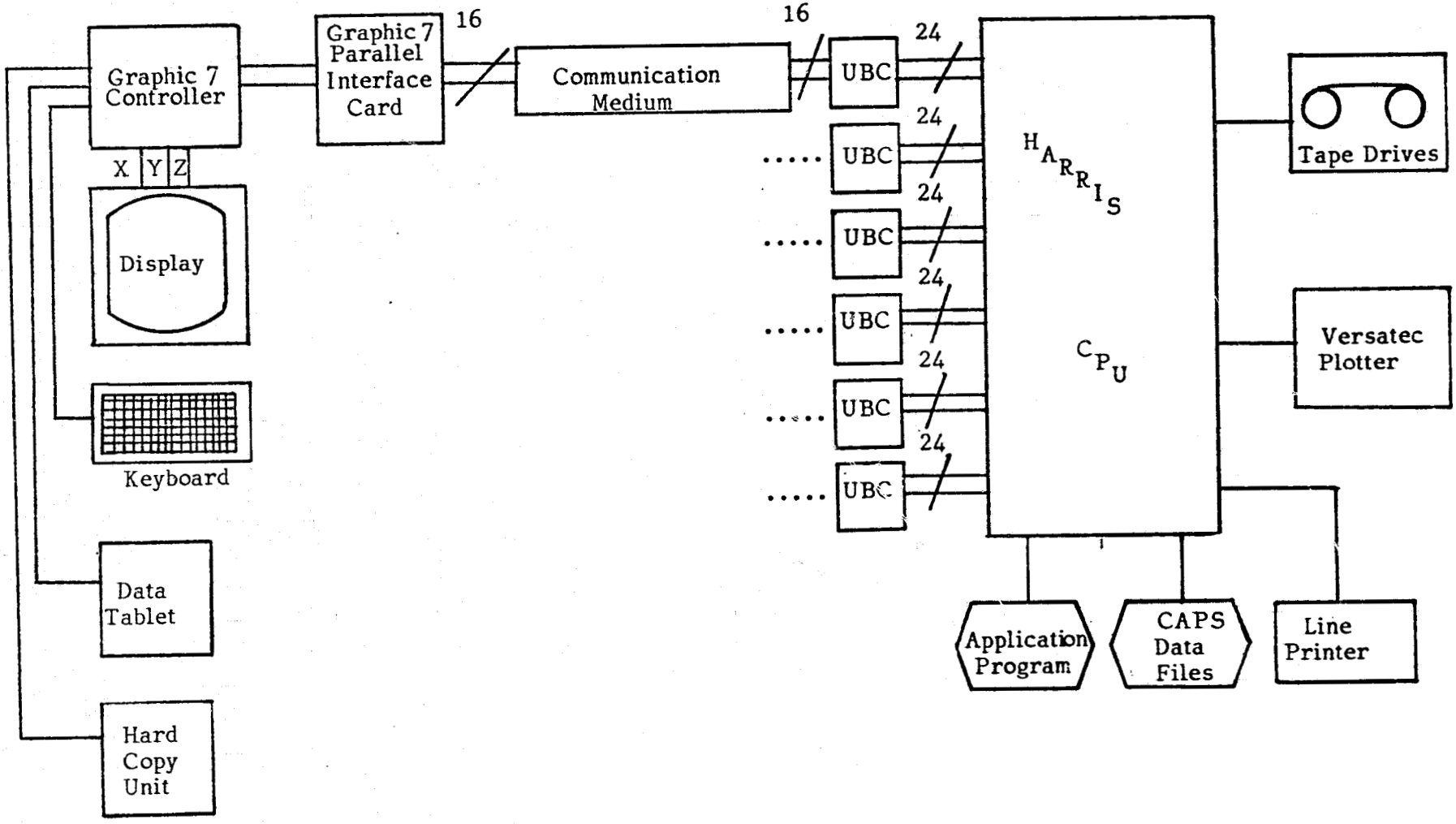


Figure 2.1-1: Block Diagram of the Crew Activity Planning System

Graphic 7, and various subsidiary functions necessary to the operation of the system.

2.1.1.1 Application Software Operation. Functions performed by the application software used on the CAPS project are discussed in documentation to be provided by the system developer. The standard Harris VULCAN operating system is used for Job Control. CAPS users from the Crew Training and Procedures Division (CTPD) have no direct control over the operation of the application software package except to the extent that certain planned responses generate requests for activation of certain software modules; that is, certain inputs which request an output from the Harris will activate subroutines to generate the appropriate display.

2.1.1.2 File Management. Several data bases are used to supply basic information for operation of the CAPS functions. These data bases are maintained by special application software again developed by McDonnell Douglas Corporation and to be documented at a later date. The user has no direct interface with this software, and the file management functions, while having a major impact on user response timing, have little or no impact on the communications interface design.

2.1.1.3 Data Conversion. The Sanders-supplied FORTRAN Support Program (FSP) [3] is operational in the Harris during CAPS activity and provides the FORTRAN interface [4] to the LSI-11 control program in the Graphic 7 controller (See Figure 2.1-2). Included in the FSP are special routines (MSGIN and MSGOUT) to process data for transmission to the Graphic 7 and insure input from the Graphic 7 is in a format recognizable by the FSP. These data conversion functions vary depending on the type of interface used to the Graphic 7. For parallel operations the characters are sent two per word. For serial interfaces the data words are communicated one byte at a time. A more detailed discussion of this data conversion is presented in Section VI.

2.2 Graphic 7 Display System

The Sanders Associates Graphic 7 display system [4] uses the improved Sanders Graphic Control Program (GCP+) system, which is contained in read only memory (ROM) in the Sanders control unit, to provide the user with a flexible, versatile

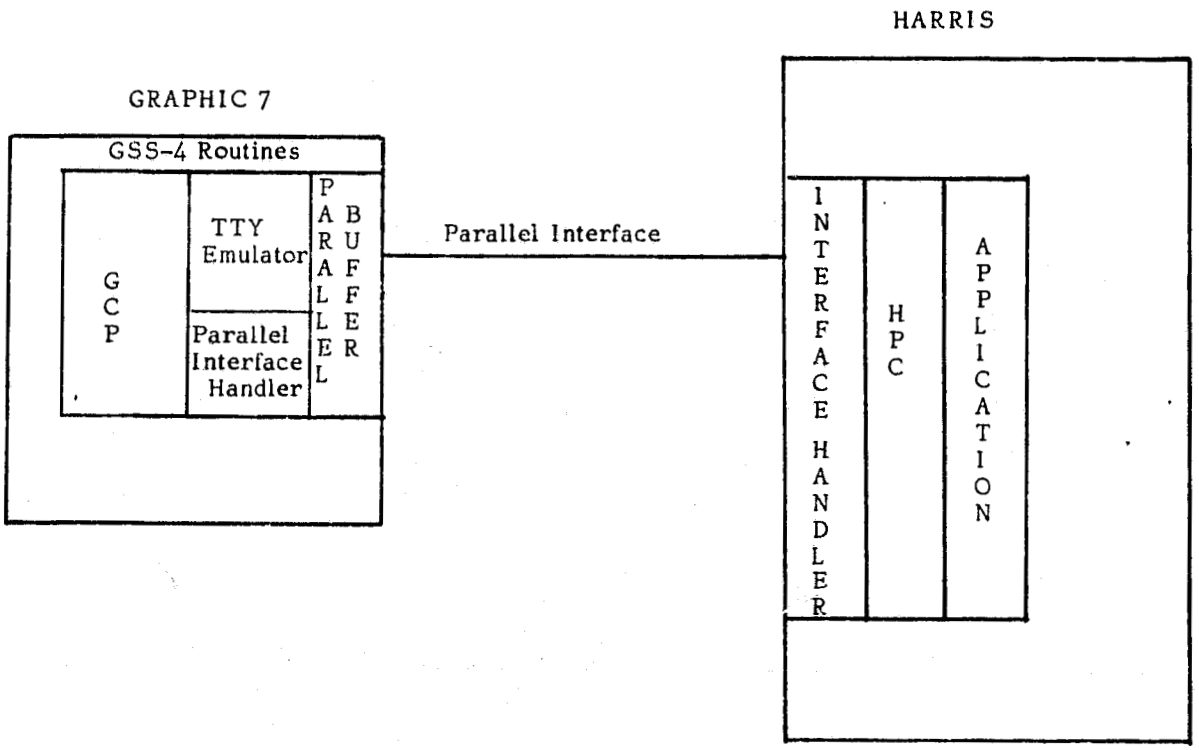


Figure 2.1-2: Parallel Interface Software Block Diagram

graphics package capable of interfacing with a large number of host computers. The developer uses standard FORTRAN subroutine calls in a program executing on the Harris host to interface with the Graphic 7 GCP+ [5]. The basic system includes support for data entry via the TALOS Data Tablet in addition to the other standard entry devices. The CAPS implementation supports entry of data from keyboard, data tablet and host computer and output to screen, Sanders Model 570 hard copy unit, Versatek Printer/Plotter, FR80 microfilm tape and host computer. The interface to the host computer is via a high speed parallel port using sixteen unidirectional data lines for input and sixteen unidirectional data lines for output as well as several special purpose signal lines to allow full or half duplex fully handshaken operations.

2.2.1 Graphic 7 Functions

The primary functions of the Graphic 7 display system are twofold:

(1) to display data presented by the host computer system and

(2) to make inputs to the host computer to allow modification of the display and data files.

Two modes of operation (described in the following paragraphs) are used by the operator of the Graphic 7 display station. The first mode is through the teletype emulator to allow use of the Graphic 7 as a TTY terminal. The second mode is used by the CTPD personnel to obtain direct access to the host application programs through the GCP+ handler in the Graphic 7 and is called system mode. Both modes can interface to the host over either a serial channel or parallel channel. In the CAPS application, parallel data communication is used to enhance data transfer speeds.

2.2.1.1 TTY mode. The TTY mode of operation is used by the system developers to have the Graphic 7 serve as an interactive device and gain access to the host at the job control level. Communication to and from the host is through a teletype emulator routine which assumes valid data characters are received one at a time in the low-order 8 bits of the data word [6]. The high order 8 bits of every data word

in this mode are ignored. Special control characters such as carriage return and line feed are recognized and acted upon accordingly.

2.2.1.2 System mode. This mode is used to allow direct communication to and from the GCP+ in the Sanders controller either serially or in parallel. In the parallel mode, all 16 bits of each data word are examined for valid system format and operated on accordingly. This is the only mode employed by the CAPS users to modify data in the system. The mode can also be used by the developers to monitor various Graphic 7 functions by using internal diagnostic procedures. Elements of the GCP+ software related to the communications interface are discussed in more detail in Section VI.

2.3 Interface Protocol

The Harris host computer communicates to the Graphics 7 terminal via a parallel interface through the Harris Universal Block Controller and the standard Sanders parallel interface board modified for a Harris interface to the GCP operating in the Graphic 7. A special handler was developed for the Harris interface by NASA's Engineering and Special Development Branch (FD7 - See [7]) to allow operation of the interface in a DMA mode or a TTY mode; that is, the interface operates in a parallel direct memory access mode or in a parallel-through-the-registers mode.

2.3.1 Data Communication

Data to be displayed on the Graphic 7 terminal can be processed by the handler in the Harris and sent to the UBC for transmission either in the DMA mode or the TTY mode. If a DMA operation is specified, the first data word is preceded by a command word to prepare the receiving unit for the DMA operation. In the CAPS implementation this command data word is ignored. All transmissions to the Graphic 7 are treated as parallel-through-the-register I/O when not in the system mode and DMA in the system mode only after a key set of register I/O operations have preceded it. No purely DMA operation takes place and the command word is never used. When the Harris operates in a DMA mode, data words are transfered directly to the UBC from memory according to the starting address and word count provided to the UBC. In the TTY mode, each data word is transmitted one byte at a time over the

channel with appropriate carriage control characters (carriage return, CNTL-E, CNTL-S, etc.) signaling the end of transmission.

2.3.2 Harris Output Data

Before data is sent to the Graphic 7, it is processed by a message routine internal to the CAPS HCP. This routine (MSGOUT) insures that minimum data is transmitted to the Graphic 7 in the minimum number of I/O operations by packing the refresh code (GCP+ inline code used to control the Graphic 7 display) generated by the standard FORTRAN code into an output buffer with contiguous memory addresses. This procedure is fully documented and discussed in [8].

2.3.3 Graphic 7 Input Data

All data transmitted to the Harris FSP by the Sanders program is in a format recognizable by the special FSP handler. The data is sent in either the TTY mode or the system mode using either the parallel-through-the-register technique or a DMA method. When sent in system mode, each word contains two bytes and is presented to the Harris UBC in that manner. In the TTY mode, each word contains one data byte and one null byte. The valid data byte is presented to the UBC in the low-order 8 bits of the parallel interface data bus.

SECTION III

BASIC CABLE COMMUNICATIONS SYSTEM

3.0 INTRODUCTION

The concept of a bus communications system was introduced to allow for flexible alteration of communication architectures, and to reduce the cost of connecting many users to a common device as well as to each other via individual direct connections. The use of bus technology allows each user to connect to a common communication medium at any one of many possible locations with minimal reconnection costs. Many types of bus architectures are employed to accomplish the desired connectivity (e.g. ring, tree, loop, star, etc.). The architecture described in this document is a modified loop architecture employing a transmission leg and a reception leg with each subscriber device connected in parallel to the two legs of the bus and the two legs connected at an element known as the "Headend."

Many bus communications networks use a time slotted protocol to allow users to transmit data over the communication medium. This type of network uses fixed or variable windows or slots to insure that data bytes are not lost due to collisions or conflicts with other users. Each subscriber is assigned a given number of time slots either at network definition time or when the user signs onto the network. Those slots not used by the subscriber are not available to any other user and thus wasted. This technique is known as Time Division Multiple Access (TDMA) and is particularly useful when users are evenly scaled, that is, the percentage of time any one user needs the network to transmit data is high in relation to the amount of time allocated to him for transmission and this percentage is nearly equal for all other user's.

When users of a communications network employ short "bursty" transmissions, that is, not well scaled, the TDMA type of network loses much of its efficiency. A different type of protocol lends itself well to this situation because all of the communications bandwidth is available to any user at any given instance on a first-come-first-served basis. In this type of protocol the users contend for the network and must insure that no other user is currently transmitting on the bus when they have data to be transmitted. One way to

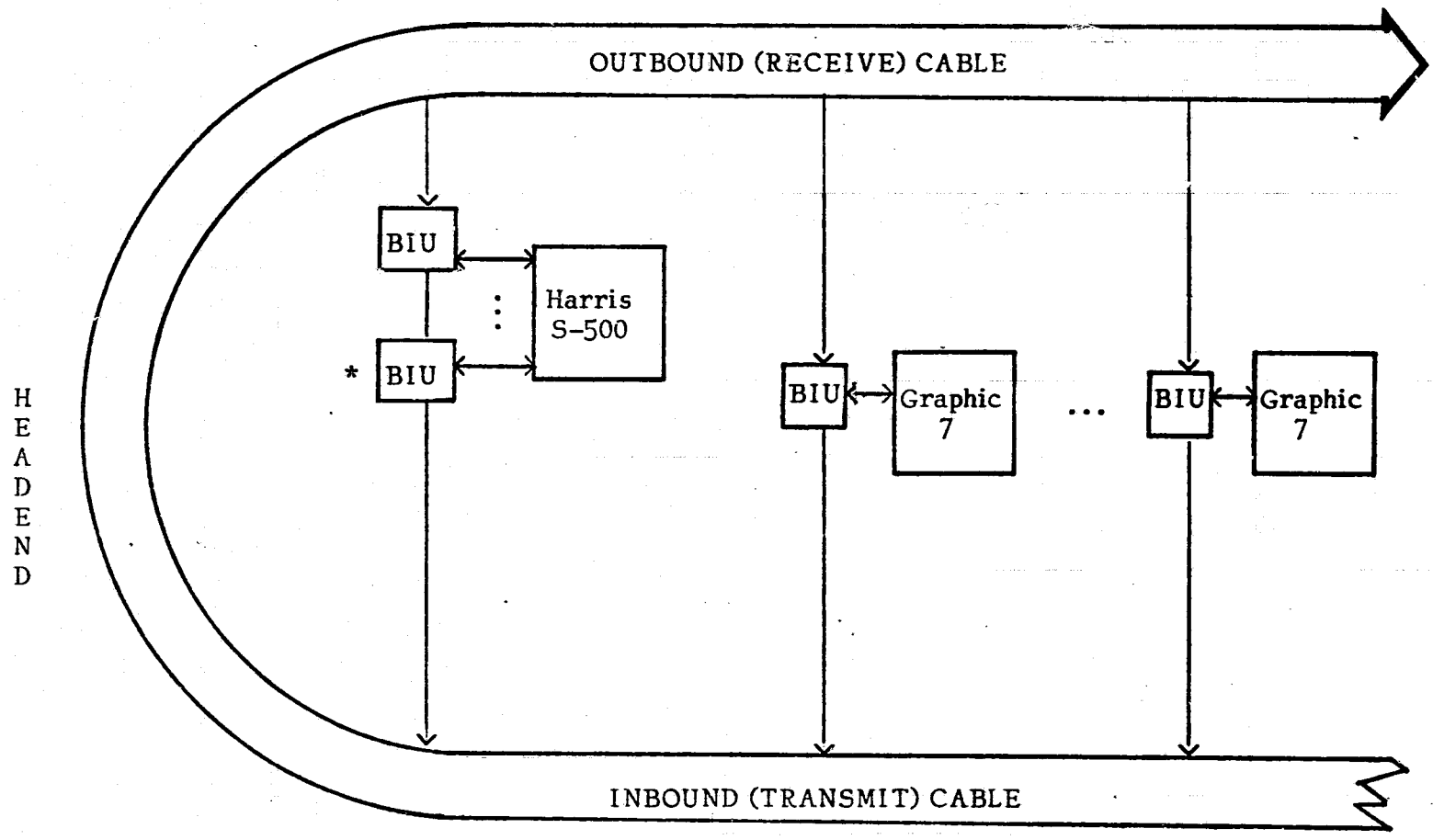
insure that no other subscriber is transmitting is to monitor the network for the presence of a transmission carrier signal. If no carrier is present, the subscriber with data to transmit assumes the network is available, raises its carrier to seize the medium, and then begins transmission of the data. It is possible for another unit to begin transmitting during the propagation delay for signals in the communications medium used. This would be the time required for the transmitting unit's carrier to reach the farthest unit on the network. For this reason each subscriber must monitor its own transmitted data for as long as it takes the data to reach the last subscriber. Collisions are detected when the transmitted data bytes are received in a garbled state. The offending subscribers then cease transmitting and wait a pseudo-random period of time before trying to retransmit the data which was previously garbled. The pseudo-random delay is used to insure that the two offending BIU's do not encounter a deadlock situation which might occur if a constant waiting period were used. The name used to describe this type of system is Carrier Sense, Multiple Access, Contention Listen-While-Talk (CSMA-LWT) and the protocol is discussed in more detail in [9] and [10].

The following paragraphs describe the NASA CAPS bus architecture and basic system operation which allows a transparent connection of the Harris UBC to the Graphic 7 parallel interface board. This communications system serves as one type of communications medium represented in Figure 2.1-1.

3.1 Bus Architecture

The NASA CAPS bus communications system is an unslotted (not using fixed subscriber time slots) CSMA-LWT system. The communications medium is a coaxial cable using standard CATV hardware. It is driven by RF modulators located in each BIU which generate a radio frequency (RF) signal within the 5 Megahertz (MHz) to 300 MHz band that is boosted when necessary by line extender amplifiers. The system is composed of a pair of main trunk cables (one outbound from and one inbound to the system headend) and several messenger or "drop" cables connected to the trunk cable by directional couplers and splitters. The center carrier frequency of the CAPS system is 24.5 MHz. For more information on the cable hardware the reader is directed to [11] and [12].

The diagram depicted in Figure 3.1-1 is a functional representation of the layout of the parts of the CAPS cable



*Note: One Harris BIU is connected to a UBC port for each Graphic 7 attached through the Bus Communications system.

Figure 3.1-1: Block Diagram of the NASA CAPS Bus Communications System

communications system. The elements discussed in this document are the Bus Interface Units (BIU) used to connect the cable system to the terminal devices and host computer.

3.1.1 Cable System Function

The two cables which constitute the communications medium provide the link between the bus interface units of the subscribers to the CAPS network. The inbound cable is used for transmission and the outbound cable is used for reception. The headend, located in a 19" rack in Room 274 of Building 4 collocated with the Harris S-500 host, is composed of a pair of line extender amplifiers and an amplifier power supply and power combiner. AC power is supplied to the line amplifiers over the communications cable through the power combiner. Figure 3.1-2 is a diagram of the headend elements of the cable system and how they are connected and provides a reference which is helpful in diagnosing communications problems, when necessary (See Section X). The headend serves as the connection point of the inbound and outbound cables as well as the point where power is supplied to the bus.

3.1.2 Bus Interface Unit Functions

The BIU's main purpose is to provide a subscriber device with a port to the high speed communication medium. The CAPS BIU is an intelligent RF modem which performs three main functions in fulfilling this purpose. These functions are:

(1) collecting subscriber data for output,

(2) processing network data packets for input to the subscriber, and

(3) gathering statistics concerning the operation of the network.

The hardware used in the construction of the all CAPS BIUs is detailed in Section V. The software used to defined the special function of the common hardware is described in Sections VII, VIII, and IX. The following paragraphs outline these three functions of the CAPS BIU.

3.1.2.1 Output Data Processing. To process output data, the BIU must gather data from the subscriber, packetize

NOTE: The arrows on the taps do not indicate direction of data flow. The tail of the arrow indicates the tap's single input/output and the head indicates the multiple output/input.

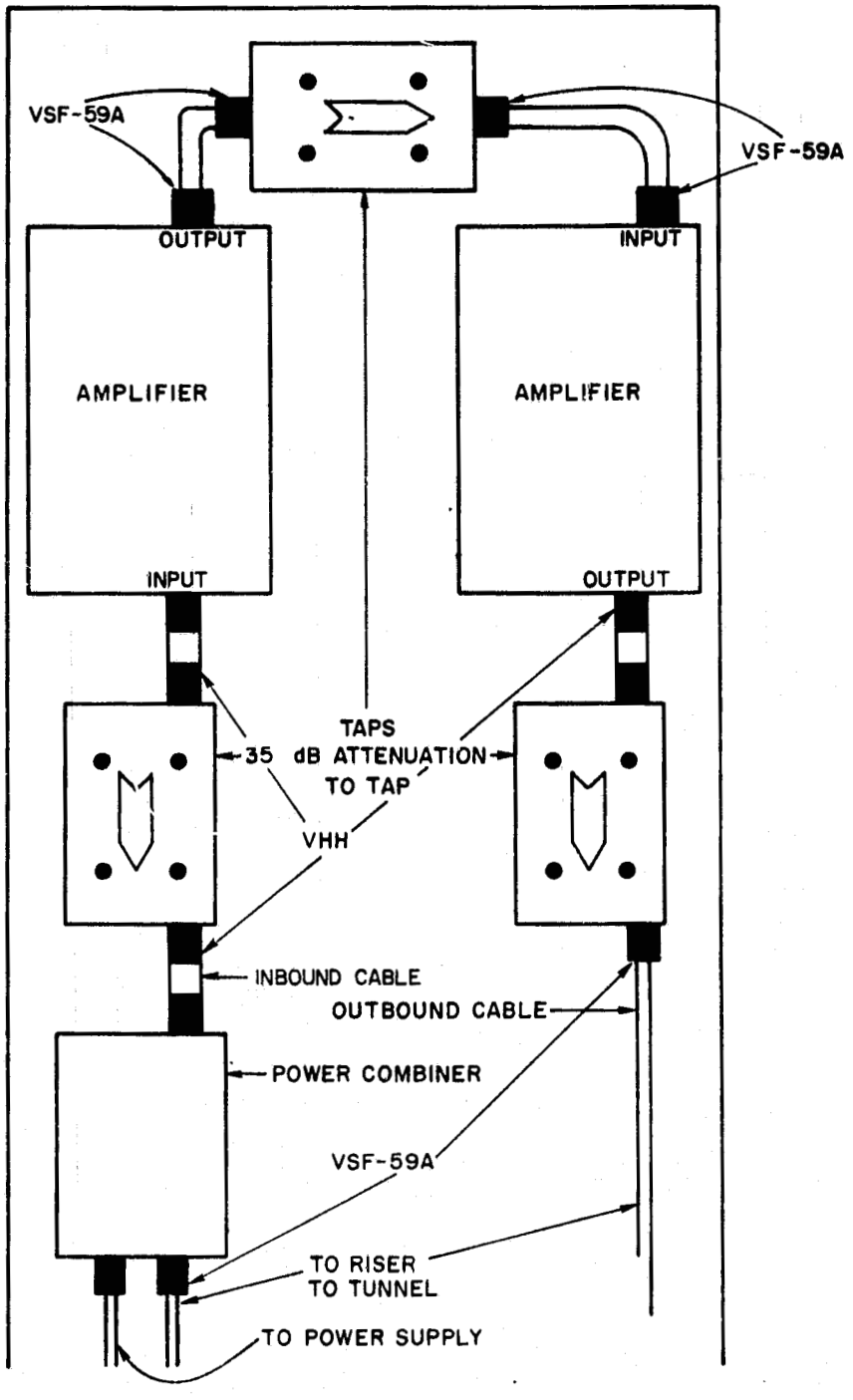


Figure 3.1-2: CAPS Headend Diagram

it, and buffer the data packets until the medium is available for transmission. In the parallel application of the CAPS system, the data interface with the subscriber is fully handshaken to insure that no data is lost when the BIU is busy and not able to process the output data as it is presented. The BIU prepares to buffer output data by adding the data words to a data packet. Each data packet is then entered in a queue of output packets awaiting transmission on the network. The output data is thus buffered until the BIU determines that the network is free for transmission (CSMA-LWT). The BIU transmits the packetized data, when possible, along with appropriate header information to allow a certain degree of error checking and correction.

3.1.2.2 Input Data Processing. The BIU receives data packets from the network based on a destination address contained in the packet header. If this destination address indicates that the subscriber device should process the packet, the data bytes are stored in an input buffer which is added to a queue of input packets awaiting processing by the subscriber. Like the output interface, the input data interface to the subscriber device is fully handshaken. Again this feature is provided to insure that no data words are lost when the subscriber is busy and not able to process the words as they are presented. This possible delay in processing input data is the main reason for the input queue. During the reception of the incoming data, the BIU completes the error checking and correction scheme cited above.

3.1.2.3 Statistics Processing. As part of the functions of the BIU, certain statistics concerning the operation of the unit are periodically transmitted on the network. These special status packets are discussed further in Paragraph 3.4.

3.2 Bus Protocol

Each device which is part of the network is connected to the cable through a BIU. These BIUs are generally designed to support a particular type of device but maintain several similar features. The common features are usually associated with the cable system interface while the unique features include the subscriber interface hardware and software. The

functions described in the following paragraphs deal with the common elements of the network protocol used between the BIUs on the CAPS cable communications system.

3.2.1 Connecting to a Valid Device

When a subscriber has data to be transmitted to another device on the network, a logical connection is established to the other device's BIU by transmitting a sign-on request. Each BIU maintains a list of the valid addresses to which it may connect. BIUs attached to terminal type devices allow the user to specify the code for the destination device. BIUs connected to general network devices such as host CPUs, "backboards" (see diagnostic procedures in Part IV), printers, etc. normally (depending on the unit's application program) respond to any BIU wishing to sign-on but rarely initiate a sign-on request of their own. If the receiving device is not currently connected to another subscriber, or if the device's software dictates that it can logically communicate with multiple subscribers, a sign-on acknowledgement packet is returned to the requester to complete the connection. If the receiver was already connected and cannot respond to multiple users, the BIU ignores the sign-on request resulting in an assumption by the requester that that particular unit is not available for a connection. Depending on the application, if a sign-on ACK packet is not received, the requesting BIU generates a message to the subscriber device asking for new information for another sign-on request or automatically generates a new sign-on request to the next unit in line.

3.2.2 Network Data Communication

With a connection established, data may be transferred in either direction in accordance with the format depicted in the example packet shown in Figure 3.2-1. The data communication discussion is divided into two parts; the packet format and the packet protocol.

3.2.2.1 Packet Format. Data bytes are transmitted on the cable in packetized form with a maximum of 120 data bytes in any one packet. An 8-byte header contains addressing information, continuity data, message type and a count of the number of bytes in the packet.

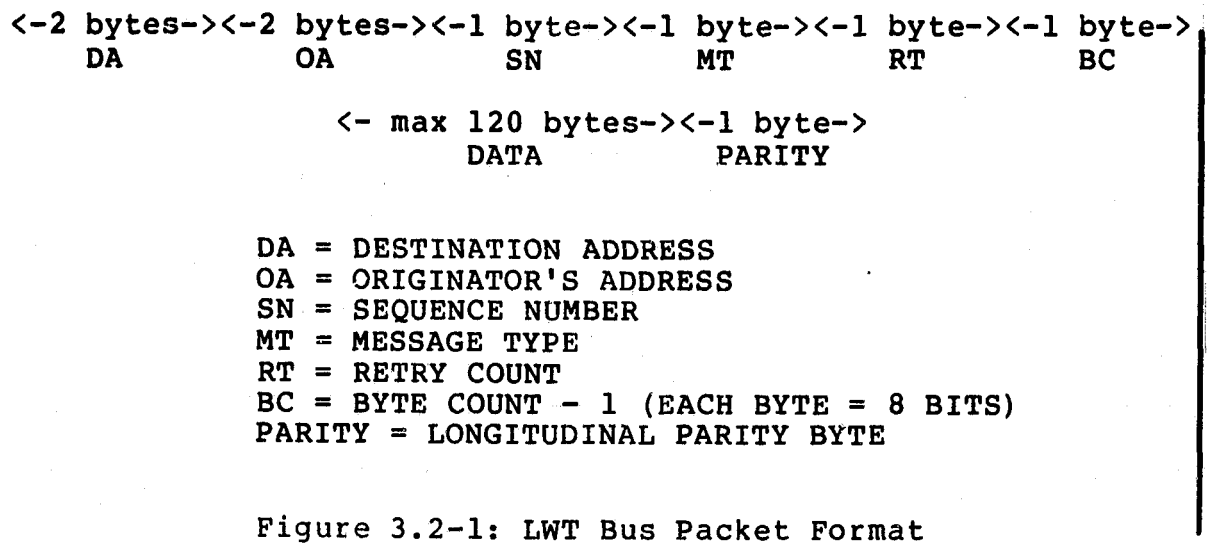


Figure 3.2-1: LWT Bus Packet Format

3.2.2.1.1 Address Information. Each subscriber on the network is known to the rest of the network by a two-byte address. As in most current applications of this type, the NASA CAPS network uses only the first byte of the address. When the receive data buffer of the BIU's receive modem becomes full, it interrupts the BIU program and thus causes the data in the buffer to be examined. If this byte is one of the first two bytes of a packet to be detected, it is assumed to be addressing information and the byte is compared against the unit's "home" address. If the packet is addressed to this unit, the packet is placed in an input buffer as it is received. If the packet is not addressed to this BIU, the receiver is turned off until the BIU's hardware detects the transmission carrier falling; that is, the transmitting BIU has finished sending the packet. At that point, an interrupt allows the BIU to reset the receiver for the receipt of the next address byte.

3.2.2.1.2 Sequence Number. Each packet transmitted on the network contains a sequence value generated by the packet originator. This value, which is incremented on each successfully transmitted packet, is useful in reconstructing data streams that are longer than the maximum allowable packet size and discarding duplicate packets. The receiving BIU stores the value of the last sequence number received and the address of the last unit to send a packet. It then uses these values to determine if any data packets have been lost or detected more than once. If it is determined that a pack-

et was lost, appropriate action can be taken depending on the application. If a packet is received more than once, it is discarded.

3.2.2.1.3 Message Type. Each bus system employing this type of protocol has a table of valid message types. Special messages are used by a given application for such things as sign-on, sign-off, reporting status, maintaining continuity and sending data. This field of the header is used to define these special message types and to direct special handling by the receiving BIU. Table I is a list of the valid message types used by the CAPS BIUs.

TABLE I

CAPS BIU MESSAGE TYPES

CODE	DEFINITION
00	DATA MESSAGE (00-1F)
DB	STATUS MESSAGE
DE	SIGN-OFF MESSAGE
DF	SIGN-ON ACKNOWLEDGMENT
E0	SIGN-ON MESSAGE (E0-FF)

NOTE: Values in parenthesis indicate valid range. In CAPS only 00 and E0 are used.

3.2.2.1.4 Retry Count. Whenever the sending BIU fails to receive an acknowledgment that the receiving BIU accepted the given packet, the sender tries to retransmit the packet at a later time. This creates a situation where the same packet may be detected more than once by a receiving BIU. The retry count is used by the the sender as a temporary memory location to keep track of the number of times a given packet has been transmitted. The value is placed in the packet header for those applications which require the receiving device to use the information to reconstruct messages which arrive out of order or to indicate that the packet was previously received. In the CAPS application, the byte is only used by the transmitting BIU.

3.2.2.1.5 Byte Count. Packets on the network can vary in length from 8 bytes (header only) to 128 bytes. To avoid

a large amount of unnecessary processing on short packets, the byte count minus one is included in the message header. This enables the receiving BIU to determine when a packet has been received completely, process the linear parity byte and return the acknowledgment if all parity checks are correct.

3.2.2.1.6 Parity. The receiving BIU detects errors in the transmission of a data packet by examining both the individual byte parity bits and the linear parity byte transmitted at the end of each packet. The transmitting BIU forms this last byte by exclusive ORing all bytes of the packet as they are transmitted. By performing the same operation on each byte as it is received, the receiving BIU has an immediate indication of any garbled data during the receipt of the packet. If the byte received at the end of the packet does not equal the parity byte calculated during reception or if any of the individual byte parity bits are found in error, the packet is ignored and no acknowledgement is sent. This causes the transmitting BIU to retransmit the packet (See Paragraph 3.2.2.2.1).

3.2.2.2 Packet Protocol. There are two elements to the packet transmission protocol: error detection and retransmission, and flow control.

3.2.2.2.1 Packet Error Handling. Packet error detection and retransmission are provided by parity checks. Each data byte contains a parity bit and each packet contains one byte of linear parity. Retransmission of a packet is generated when a transmitting subscriber waits for approximately 100 usecs for an acknowledgement of the packet by the addressee BIU. If no acknowledgement is received, the transmitting unit assumes that the packet did not arrive at the receiving BIU with good parity checks and the packet is retransmitted a number of times and then discarded.

3.2.2.2.2 Flow Control. Due to the limited buffer space available in each BIU, it is possible for a given unit to become saturated with data packets either awaiting transmission on the network or processing by the subscriber device. The flow control feature of the protocol helps to eliminate the loss of data if no buffers are available at any given time. While this condition exists, the receiving BIU acknowledges packets with a special character (See Paragraph

7.3.2) that informs the transmitting BIU of its buffer overflow condition. The transmitting BIU waits for a specified period of time before retransmitting the packet thus allowing the receiving BIU time to clear one or more buffers via normal processing.

3.2.3 Subscriber Data Communication

The subscriber interface is usually unique to the attached device. Each BIU, while using a common interface (both hardware and software) to the cable communications system, has a special interface (software and sometimes hardware) with its subscriber. For the most part, these special requirements are handled in the firmware program in the BIU's Programmable Read Only Memory (PROM). The interface hardware can be flexible enough to allow a BIU to interface to several largely different devices depending on the cable connectors used on the BIU chassis. In the CAPS application, the hardware configuration is identical for both the Harris BIU and the Graphic 7 BIU, while the PROM programs allow the Harris BIU to resemble a Graphic 7 display to the Harris CPU and the Graphic 7 BIU to respond like the Harris UBC to the Graphic 7 parallel interface.

3.2.4 Status Recording

As part of the BIU's responsibility, certain statistics concerning the operation of the BIU are maintained and periodically transmitted to a special network address (See Paragraph 7.9 and Table II) for processing by some type of status recording device. Statistics gathered include such things as the number of packets each BIU transmitted successfully, the number of packets received with good parity and the number of packets received with bad parity. All the data bytes contained in the status packet are detailed in Table II below.

The primary use of these packets is to determine the operational performance of the network and allow for analysis of problems that might occur in one or more of the BIUs. An example of how this feature can be used is as follows: when a BIU's receiver fails or is marginal, the number of packets received with good parity will decrease and the number of discarded packets will increase since this BIU will not receive messages addressed to it or the acknowledgments of the data packets transmitted by it. For further information on this subject the reader is directed to Part IV of this report and to [13].

TABLE II

STATUS PACKET DEFINITION

HEX BYTE #	DEFINITION
00-01	DESTINATION ADDRESS (00)
02-03	ORIGINATOR ADDRESS
04	SEQUENCE NUMBER
05	MESSAGE TYPE (DB)
06	RETRY COUNT
07	BYTE COUNT (29)
08-09	NUMBER OF TRANSMITTED PACKETS
0A-0B	NUMBER OF RETRANSMITTED PACKETS
0C-0D	NUMBER OF COLLISIONS DETECTED
0E-0F	NUMBER OF DISCARDED PACKETS
10-11	NUMBER OF RECEIVED PACKETS (GOOD)
12-13	NUMBER OF RECEIVED PACKETS (BAD)
14-15	NUMBER OF PACKETS LOST (NO BUFFER)
16-17	NUMBER OF TIMES BIU FOUND BUSY BUS
18	NUMBER OF OUTPUT PACKETS WAITING
19	NUMBER OF INPUT PACKETS WAITING
1A	NETWORK UART STATUS
1B	DEVICE UART STATUS
1C	PARALLEL PORT 1 INTERRUPT STATUS
1D	CURRENT TRANSMITTING ADDRESS
1E-29	12-BYTE FUNCTION DESCRIPTION

PART II

SYSTEM HARDWARE

SECTION IV

CAPS COMMUNICATIONS HARDWARE INTERFACE

4.0 INTRODUCTION

To allow for an understanding of the functioning of the hardware used by the CAPS system and how it interfaces with the BIU, a limited discussion of the Harris UBC and Sanders parallel port is presented in this section. Data lines, handshaking signals and various other control signals are described in detail. Further details on the two interfaces can be found in [14], [15] and [16].

4.1 Harris UBC Interface

The Harris Universal Block Controller is designed to interface with magnetic tape drives, disk controllers, and any other type of high speed parallel device needing a fully handshaken interface with the Harris Slash 8 CPU. All interface handshaking signals are presented as levels and the connected device's returned signals are buffered for processing. A word count zero interrupt line to the CPU interrupt logic board is provided to allow for full DMA operation. In addition, an interrupt line from each device connected to the UBC is passed directly to the Harris interrupt logic to allow for input interruption of the CPU.

4.1.1 Data Lines

In the Harris UBC, 24 buffered data lines are provided for input (DFU00-23) and an additional 24 lines are used for output (DTU00-23). Tri-state drivers are used on the output lines to insure that no loading of peripheral devices occurs when data is not valid. The interface pin definition is shown in Table III below. Note that for the CAPS interface only the low-order 16 bits for both input and output are utilized. Table IV is used to clarify the connections from the Harris UBC to the Graphic 7 parallel interface card when bypassing the BIUs.

TABLE III

HARRIS/GRAPHIC 7 PARALLEL INTERFACE CONTROL SIGNALS

HARRIS SIGNAL NAME	UBC PIN #	BIU SIGNAL NAME		BIU CONNECTOR PIN #	BIU BACKPLANE CONNECTOR	GRAPHIC 7 SIGNAL NAME	PARALLEL CARD PIN #
DTU00	P3-36*	INPUT	PA0A	2	EC67	ID00	P3-1*
DTU01	P3-38	DATA	PA1A	3	EC68	ID01	P3-3
DTU02	P3-40	(6522A)	PA2A	4	EC69	ID02	P3-5
DTU03	P3-42		PA3A	5	EC70	ID03	P3-7
DTU04	P3-44		PA4A	6	EC71	ID04	P3-9
DTU05	P3-46		PA5A	7	EC72	ID05	P3-11
DTU06	P3-48		PA6A	8	EC73	ID06	P3-13
DTU07	P3-50		PA7A	9	EC74	ID07	P3-15
DTU08	P2-6		PB0A	10	EC75	ID08	P3-17
DTU09	P2-8		PB1A	11	EC76	ID09	P3-19
DTU10	P2-10		PB2A	12	EC77	ID10	P3-21
DTU11	P2-12		PB3A	13	EC78	ID11	P3-23
DTU12	P2-14		PB4A	14	EC79	ID12	P3-25
DTU13	P2-16		PB5A	15	EC80	ID13	P3-27
DTU14	P2-18		PB6A	16	EC81	ID14	P3-29
DTU15	P2-20		PB7A	17	EC82	ID15	P3-31
CDH	P3-32	CB1A		20	EC84	IIFU	P3-45
ODH	P3-34	CA1A		19	EC83	IWR	P3-41
ODACP	P3-8	CA2A		21	EC85	ICTL	P3-37
DISC	P3-26	CA1C		45	EC95	CNCT	P3-35
CNCT	P3-10	CA2C		46	EC96	DISC	P3-39
IIFU	**	CB2C		49	EC116	CDH	P3-43
$\overline{R}/W$	P3-72		PA5C	47	EC114	$\overline{R}/W$	P2-47

* NOTE: Both Harris UBC and Graphic 7 pin numbers are referenced to the BIU connector. To determine direct Harris UBC-to-Graphic 7 connections, refer to TABLE IV.

** Indicates direct connection to the CPU's interrupt interface.

TABLE III (Concluded)

HARRIS SIGNAL NAME	UBC PIN #	BIU SIGNAL NAME		BIU CONNECTOR PIN #	BIU BACKPLANE CONNECTOR	GRAPHIC 7 SIGNAL NAME	PARALLEL CARD PIN #
DFU00	P3-78	OUTPUT	PA0B	22	EC42	OD00	P2-1
DFU01	P3-74	DATA	PA1B	23	EC43	OD01	P2-3
DFU02	P3-70	(6522B)	PA2B	24	EC44	OD02	P2-5
DFU03	P3-66		PA3B	25	EC45	OD03	P2-7
DFU04	P3-62		PA4B	26	EC46	OD04	P2-9
DFU05	P3-58		PA5B	27	EC47	OD05	P2-11
DFU06	P3-54		PA6B	28	EC48	OD06	P2-13
DFU07	P3-52		PA7B	29	EC49	OD07	P2-15
DFU08	P2-74		PB0B	30	EC50	OD08	P2-17
DFU09	P2-72		PB1B	31	EC51	OD09	P2-19
DFU10	P2-70		PB2B	32	EC52	OD10	P2-21
DFU11	P2-68		PB3B	33	EC53	OD11	P2-23
DFU12	P2-66		PB4B	34	EC54	OD12	P2-25
DFU13	P2-64		PB5B	35	EC55	OD13	P2-27
DFU14	P2-62		PB6B	36	EC56	OD14	P2-29
DFU15	P2-60		PB7B	37	EC57	OD15	P2-31
DAVFU	P3-12	CA2B		40	EC88	OCTL	P2-41
DATU	P3-30	CA1B		39	EC87	OWR	P2-37
UR00	P3-18	UNIT	PA0C	41	EC91	UR00	P2-35
UR01	P3-20	REG.	PA1C	42	EC92	UR01	P2-39
UR02	P3-22	(6522C)	PA2C	43	EC93	UR02	P2-43
UR03	P3-24		PA3C	44	EC94	UR04	P2-45

TABLE IV

HARRIS-TO-GRAPHIC 7 INTERFACE CONNECTIONS

HARRIS SIGNAL NAME	UBC PIN #	GRAPHIC 7 PIN #	GRAPHIC 7 SIGNAL NAME
DTU00	P3-36	P2-1	OD00
DTU01	P3-38	P2-3	OD01
DTU02	P3-40	P2-5	OD02
DTU03	P3-42	P2-7	OD03
DTU04	P3-44	P2-9	OD04
DTU05	P3-46	P2-11	OD05
DTU06	P3-48	P2-13	OD06
DTU07	P3-50	P2-15	OD07
DTU08	P2-6	P2-17	OD08
DTU09	P2-8	P2-19	OD09
DTU10	P2-10	P2-21	OD10
DTU11	P2-12	P2-23	OD11
DTU12	P2-14	P2-25	OD12
DTU13	P2-16	P2-27	OD13
DTU14	P2-18	P2-29	OD14
DTU15	P2-20	P2-31	OD15
CDH	P3-32	P3-43	CDH
ODH	P3-34	P2-41	OCTL
ODACP	P3-8	P2-37	OWR
DISC	P3-26	P3-39	DISC
CNCT	P3-10	P3-35	CNCT
$\overline{\text{IIFU}}$	*	P3-45	$\overline{\text{IIFU}}$
$\overline{\text{R}}$/W	P3-72	P2-47	$\overline{\text{R}}$/W

* Indicates direct connection to the CPU's interrrupt interface

TABLE IV (Concluded)

HARRIS SIGNAL NAME	UBC PIN #	GRAPHIC 7 PIN #	GRAPHIC 7 SIGNAL NAME
DFU00	P3-78	P3-1	ID00
DFU01	P3-74	P3-3	ID01
DFU02	P3-70	P3-5	ID02
DFU03	P3-66	P3-7	ID03
DFU04	P3-62	P3-9	ID04
DFU05	P3-58	P3-11	ID05
DFU06	P3-54	P3-13	ID06
DFU07	P3-52	P3-15	ID07
DFU08	P2-74	P3-17	ID08
DFU09	P2-72	P3-19	ID09
DFU10	P2-70	P3-21	ID10
DFU11	P2-68	P3-23	ID11
DFU12	P2-66	P3-25	ID12
DFU13	P2-64	P3-27	ID13
DFU14	P2-62	P3-29	ID14
DFU15	P2-60	P3-31	ID15
DAVFU	P3-12	P3-41	IWR
DATU	P3-30	P3-37	ICTL
UR00	P3-18	P2-35	UR00
UR01	P3-20	P2-39	UR01
UR02	P3-22	P2-43	UR02
UR03	P3-24	P2-45	UR03

4.1.2 Handshake Lines

As metioned above, the interface is fully handshaken with signal levels. This is to insure that data can not be lost during normal data transfer operations; that is, both devices can accept data at their own rate without loss. The paragraphs below describe the handshaking signals and how they are used. Refer to Table III for the connector pin locations of the various handshaking signals.

4.1.2.1 Command Data Here (CDH). During DMA operations, each output command is preceded by a command data word that informs the destination device to prepare to receive an output of the specified number of words. This command word causes the device to initialize itself for the transfer by setting a word count register, if available. The UBC raises this handshake signal whenever a valid command word is present on DTU00-23. This signal is held high and the data lines remain valid until the destination device raises ODACP to signal that it has accepted the command data word. In the CAPS system, the Sanders Graphic 7 ignores the CDH signal because all output transfers are parallel-through-the-register and controlled by the ODH control signal. The Graphic 7 simply returns the CDH signal to the UBC on the ODACP line to complete the handshake process. This is discussed further in later paragraphs detailing the Graphic 7 operation.

4.1.2.2 Output Data Here (ODH). The UBC signals to the receiving device that a valid data word is present on the data lines by raising ODH. As in the case of the CDH signal the UBC holds the data lines steady and continues to hold ODH high until it receives the ODACP signal from the connected device. A rough timing diagram of this handshake process is shown in Figure 4.1-1. This diagram is valid for both the ODH and CDH signals; however, the time delay at the Graphic 7 is much shorter in the case of CDH for the CAPS implementation. The diagram also contains information on the signal timing when the BIUs are installed. This provides the reader with a method of comparison of the timing associated with local data transfers and bus communication system transfers.

4.1.2.3 Output Data Accepted (ODACP). As mentioned above, ODACP is the return handshake from the unit to indicate that the data on the DTU bus has been received. The unit must hold this signal high until the UBC drops the ODH

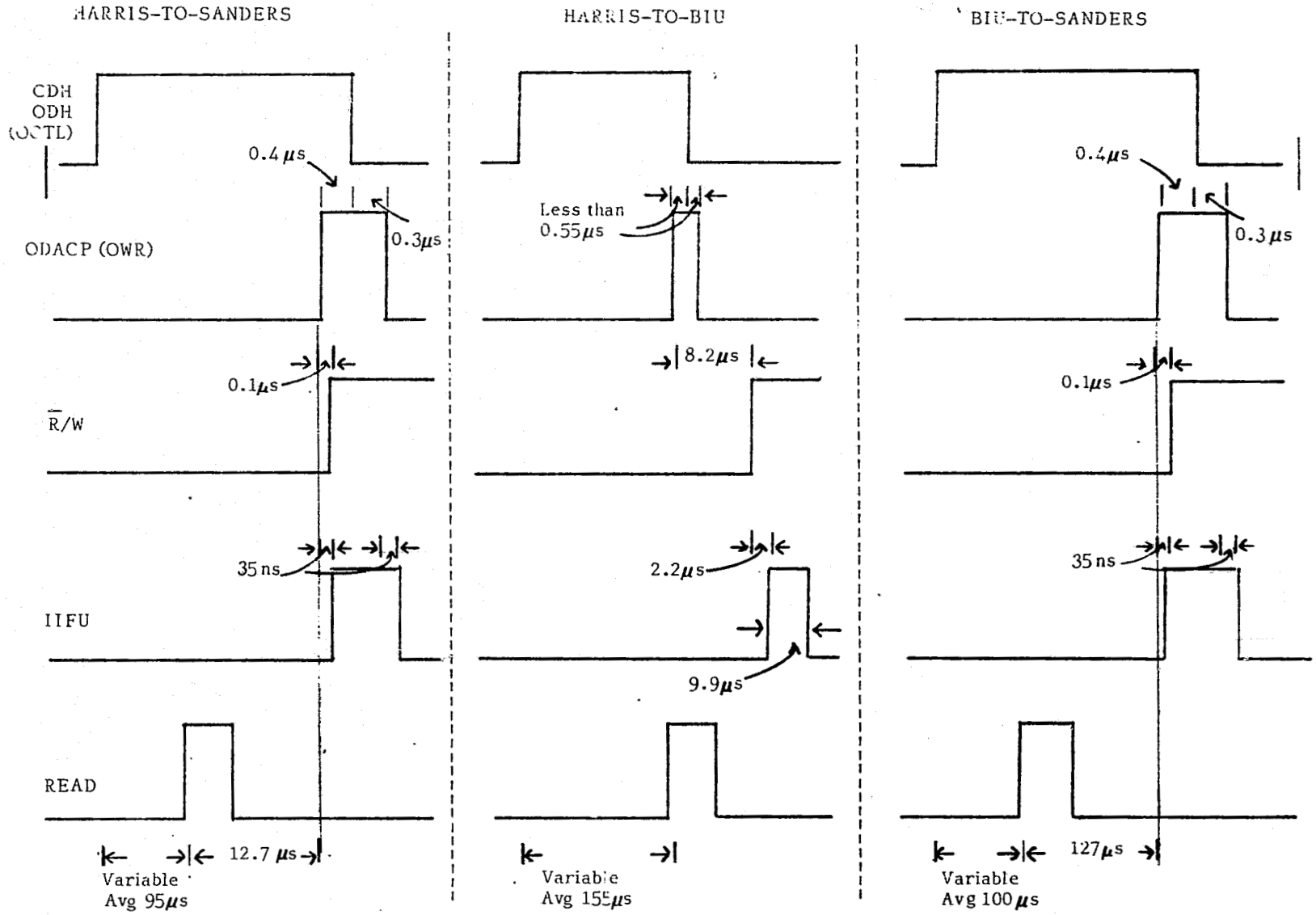


Figure 4.1-1: Harris UBC Data Output Timing

or CDH signal to insure a proper handshake sequence. The Harris UBC uses this signal to trigger the resumption of normal operations.

4.1.2.4 Data Available From Unit (DAVFU). Whenever an input word is available for the Harris UBC, the attached unit raises the DAVFU signal and interrupts the Harris via the IIFU signal line. The presence of the DAVFU signal is interpreted by the UBC to mean that valid input data is available on the DFU bus. This handshake signal is held by the unit until the UBC raises the DATU signal line to indicate that it has received the data. The timing diagram presented in Figure 4.1-2 depicts this handshaking process as well as the timing of the BIU interfaces. Again a timing comparison can be made between the local and BIU operations.

4.1.2.5 Data Accepted to Unit (DATU). As mentioned above, the DATU signal is generated by the UBC to indicate to the unit sending an input word that the data has been accepted by the UBC. As in the case of the ODACP signal, the UBC holds this signal high until the sending unit drops the DAVFU signal. The sending unit may then continue with other operations.

4.1.2.6 Input Interrupt From Unit (IIFU). The original intent of the IIFU signal was to interrupt the Harris processor whenever the external unit had data ready for input. This function is still used in the CAPS implementation; however, the signal has been given an additional function to perform. To allow the Harris communications handler to operate in a teletype (TTY) mode, this line is pulsed after each output word to provide an interrupt signal indicating to the handler that the last word sent from the UBC was accepted. This modification to the Graphic 7 parallel interface card is discussed in later paragraphs. As far as the Harris handler is concerned, therefore, this interrupt can mean either an input or output has occurred and the $\overline{R}/W$ status line must be examined to determine which type of operation was signaled by the interrupt.

4.1.2.7 $\overline{\text{READ}}/\text{WRITE}$ Status Line (SFU01). To allow for the Harris interface handler to determine the meaning of the IIFU signal, the $\overline{R}/W$ status line was implemented. As in all discussions of the CAPS parallel interface, the input/output

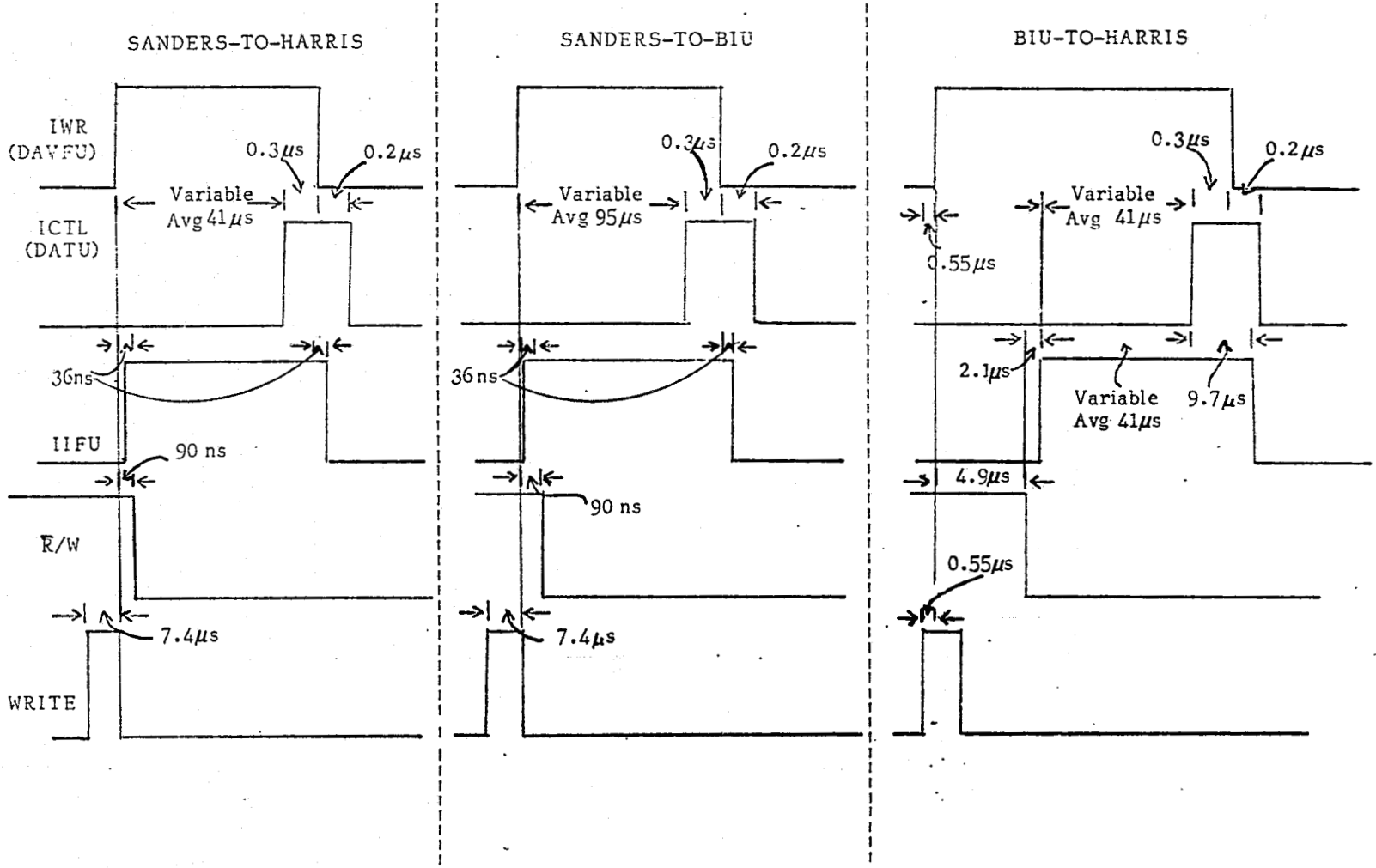


Figure 4.1-2: Harris UBC Data Input Timing

terms are referenced to the Harris host; that is, input (read) refers to data coming into the host and output (write) refers to data leaving the host. The handler uses status line 01 of the UBC status bus to determine which operation was performed last. When the Harris UBC sends a data word to the unit (write operation), the unit raises the $\overline{R}/W$ signal and holds it high until the unit has data to send to the Harris UBC. When the external unit (Graphic 7) raises its DAVFU signal line and sets the IIFU signal high (read operation), it clears the $\overline{R}/W$ signal line and holds it low until the next output word is received from the Harris UBC. (See Paragraphs 4.2.2.6 and 4.2.2.7 for more detail on the operation of this signal).

4.1.2.8 Connect Line (CNCT). The Harris UBC interface can support up to 16 peripheral devices on one channel. Since only one pair of I/O connectors exists on every UBC, multiple devices share a common signal bus via a daisy-chain cable. As a result, each device must use tri-state drivers to isolate its output stage. To control which device is communicating to the UBC at any particular instant, the CNCT line is used in conjunction with the four unit register (UR) lines and the disconnect (DISC) line. The UBC contains a unit code register (UCR) which maintains the address of the last unit involved in an I/O operation. The input or output sequence starts with a hardware interrogation of the unit code register. This allows the UBC to determine if the next I/O operation involves the same device. If it does and the CNCT signal is high, the operation proceeds normally. If the UCR disagrees with the address of the unit specified for the next I/O or the CNCT line is low, a connect/disconnect sequence is executed by the hardware.

To facilitate daisy-chaining all control and data lines, the CNCT signal is also used by the devices connected to the UBC to enable the tri-state drivers on all outgoing control and signal lines (not necessary on incoming lines). The proper operation of the connect/disconnect sequence in the BIU and the Graphic 7 is inhibited by this arrangement, since a delay of the tri-state control signal is necessary to allow the CNCT signal change of state to be registered by the UBC. To correct this hardware problem, a modification to the Graphic 7 parallel interface board is required. This modification is detailed in Paragraph 4.2.2.8 and [17].

4.1.2.9 Disconnect Line (DISC). To perform a connect/disconnect sequence the UBC raises the DISC signal. All units attached to the controller are forced to drop their CNCT signals when the DISC signal is detected. The DISC signal forces each unit to interrogate the UR lines and compare the address to its own internal address. The unit, which decodes the UR lines and finds its address, raises the CNCT signal immediately after the UBC drops the DISC signal. A timing diagram of this sequence appears in Figure 4.1-3. Once again the diagram includes BIU timing for comparison.

4.1.2.10 Unit Register Lines (UR00-03). As mentioned above, the UR lines are used by the UBC to indicate which of the 16 possible devices attached to the channel is being addressed for a particular I/O operation. In the CAPS implementation only Unit Code 0 is used since each UBC will be connected to only one Graphic 7. This was necessary because a daisy-chain connector for the Sanders Graphic 7 display does not exist and the development of the Harris parallel interface handler would result in too large a cost.

4.2 Graphic 7 Parallel Interface

Each Graphic 7 display station is composed of two 25-inch display CRT's, a control cabinet housing the digital logic for both graphics and communication operation, a TALOS data tablet, a standard alphanumeric keyboard and the various cables necessary to interconnect the components. In addition, each station shares access to a hard copy unit with one or two other stations. To allow parallel communications to and from the host, Sanders Associates provides a general parallel interface card with a specified host area that can be wired for any of several selected hosts. For the CAPS application, the parallel cards were ordered with the standard Harris interface. As is explained in the following paragraphs, several additional modifications to the standard Graphic 7-to-Harris parallel interface were necessary to allow the operation of the Sanders as a TTY terminal. In addition, to support communications through the bus interface unit, a wiring change involving the CNCT signal was necessary.

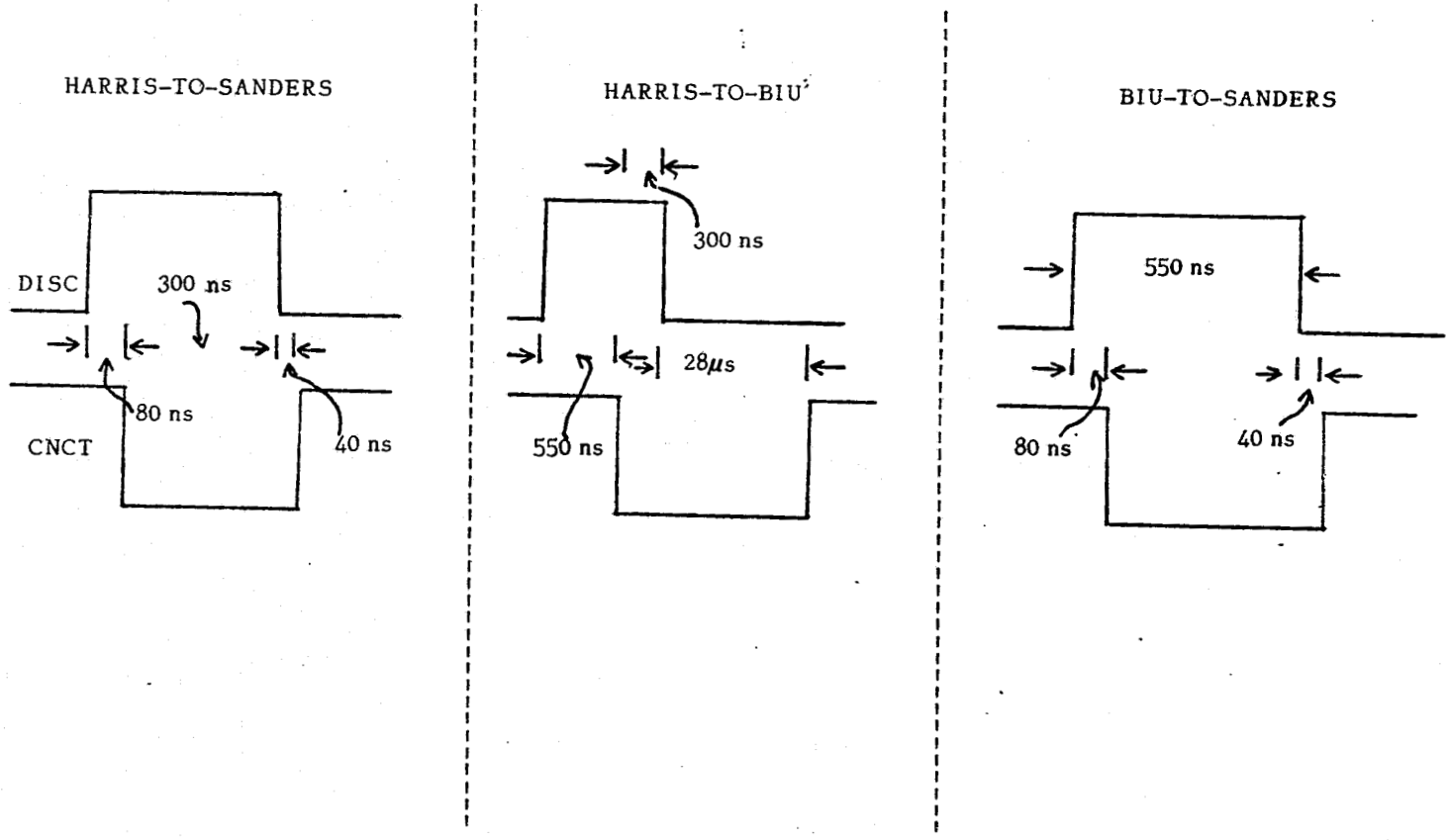


Figure 4.1-3: Harris UBC CNCT/DISC Timing

4.2.1 Data Lines

Sixteen unidirectional buffered data lines are provided by the standard interface card for the input bus (ID00-15) and 16 unidirectional lines for the output data bus (OD00-15). The output lines are connected to tri-state drivers which are controlled by the CNCT signal on the standard Harris version of the card. The interface pin definition is shown in Table III.

4.2.2 Handshake Lines

The standard Graphic 7 parallel interface card provides both handshake pulses and handshake levels. The Harris UBC modification to the standard card uses only the level handshake signals. Each signal is discussed below and its corresponding Harris definition is indicated in parentheses. Several special signals are added to the general interface card to allow for:

(1) IIFU interruption on output,

(2) $\overline{R}/W$ signal control, and

(3) CNCT signal change-of-state detection.

These signals as well as the standard Harris interface signals are shown on the schematic diagram in Figure 4.2-1. Each new signal is explained in detail in one of the following paragraphs.

4.2.2.1 Command Data Here. As mentioned above in Paragraph 4.1.2.1, this signal is generated by the Harris UBC whenever a DMA operation is initiated. The Graphic 7 parallel interface card was modified by Sanders to turn this signal around as the output data accepted signal (ODACP). For this reason, it can be seen in Figure 4.2-1 that the CDH signal drives two inputs of the four input NAND gate U77 through inverter U82E. This in turn causes the OWR signal to go high through driver U81C whenever the CDH signal is high.

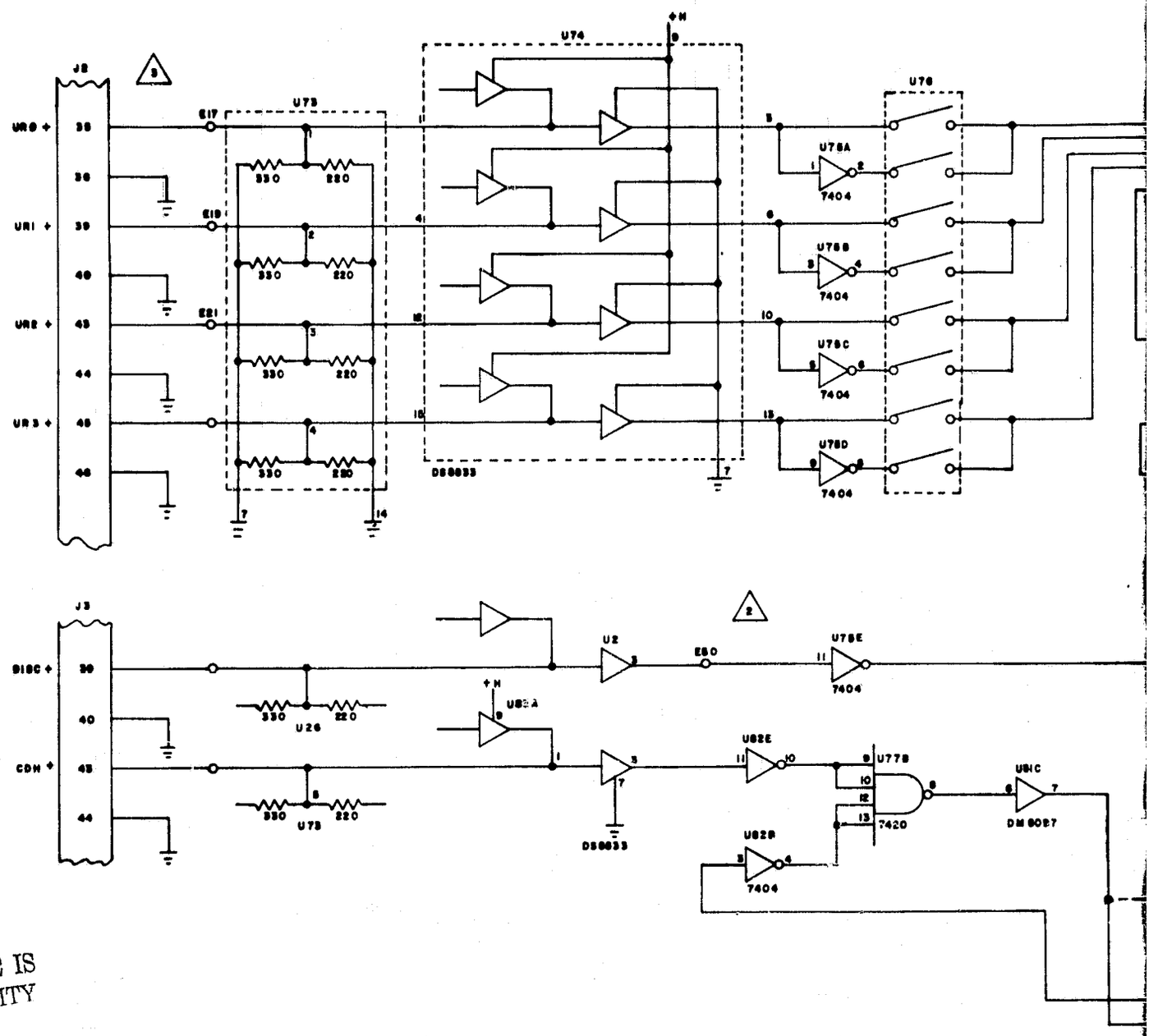
4.2.2.2 Output Control (OCTL). This signal, which is the equivalent of the Harris ODH signal, is used to indicate that the data present on the OD00-15 bus is valid output

data. The host holds this signal high until the Graphic 7 responds with the OWR handshake level. The OCTL signal gates the input data bus signal lines into the input data buffer.

4.2.2.3 Output Word Read (OWR). This signal is the equivalent of the Harris ODACP signal. The Graphic 7 uses this line to indicate that it has accepted the data on the OD bus and it is ready for the next input. The timing for all of these signals is roughly the same as that shown in Figure 4.1-1. The Graphic 7 holds this signal high until the host drops the OCTL signal. The reference to OWR in Figure 4.2-1 is only to show its relationship to the CDH signal, to the control of the $\overline{R}/W$ status line and to the signaling of an acceptance interrupt from unit. The schematic shows that in the Harris version of the parallel interface card the normal path for the OWR signal is broken at E18 and attached through inverter U82B to the other two inputs of NAND gate U77. In this way when either OWR or CDH is high, there is a high output from J2-37. To control the $\overline{R}/W$ status line and indicate when the host has issued a write command, the OWR signal is taken from driver U81C and inverted by U82F to provide the preset input to flip-flop U80B. In addition, this signal is used to drive one input of NAND gate U78D which provides the IIFU signal.

4.2.2.4 Input Word Ready (IWR). This signal is used by the Graphic 7 to indicate to the host that valid input data is available on the ID bus. It is the Harris equivalent of the DAVFU handshake signal. The signal is held high until the ICTL input goes high. The signal is passed directly through driver U66 on the standard Graphic 7 board but tapped at its output (U66-10) to provide the input for the IIFU interrupt signal. This can be seen on Figure 4.2-1.

4.2.2.5 Input Control (ICTL). This signal is used by the host to inform the Graphic 7 that the input data on the ID bus has been received. It is used to disable the IWR output and reinitialize the Graphic 7 for the next input word to the host. As can be seen in Figure 4.2-1, it is also used as the enabling signal for the IIFU output. When ICTL goes high, it turns off U78C through inverter U75F thus causing the IIFU signal to fall.

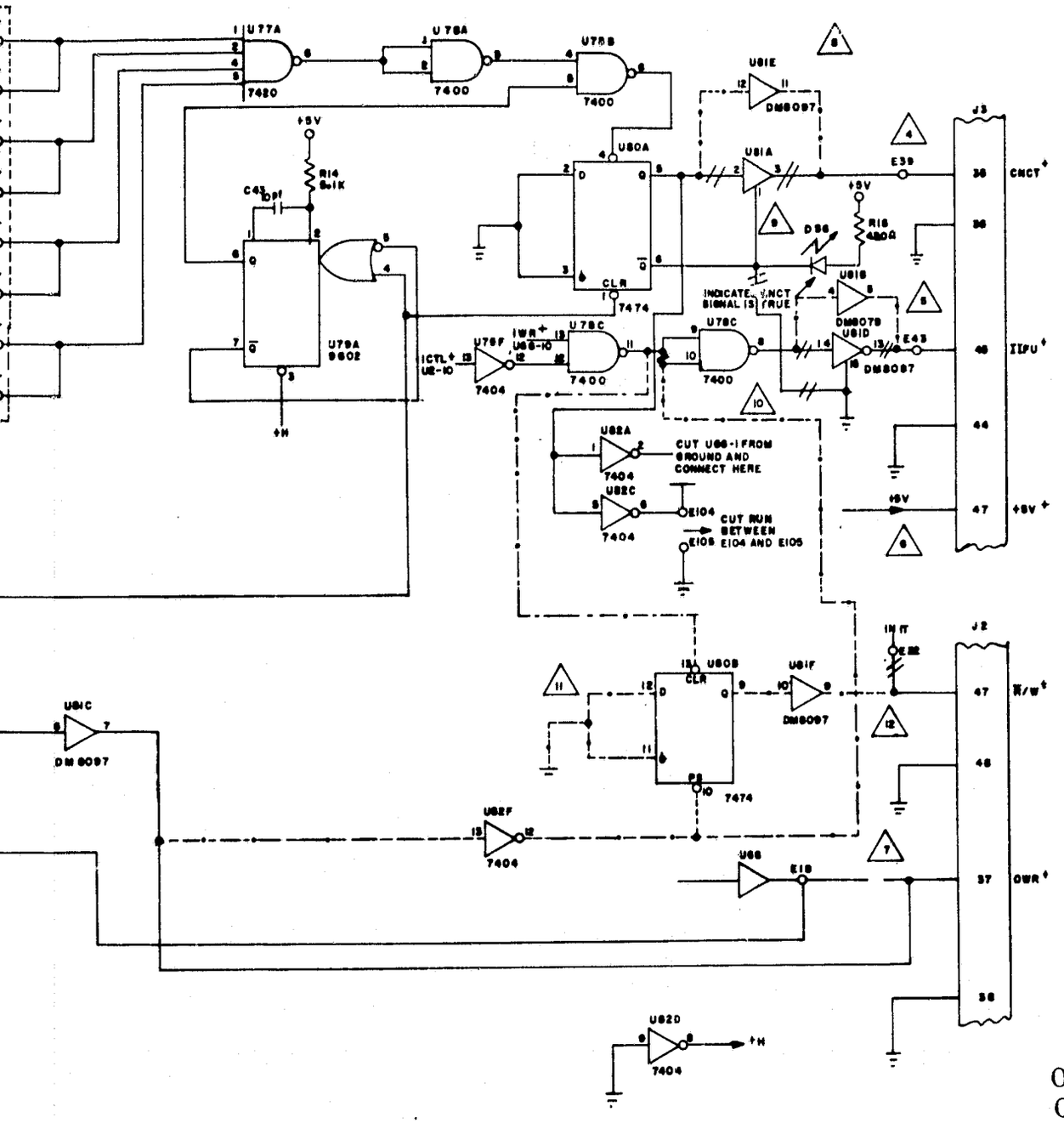


ORIGINAL PAGE IS
OF POOR QUALITY

FOLDOUT FRAME

12. CUT PC RUN BETWEEN E22 AND J2 PIN 47. CONNECT NEW CIRCUITRY AS SHOWN.
11. WIRE SPARE FLIP-FLOP U80B AS SHOWN.
10. CUT CONNECTION BETWEEN U78 PINS 9 AND 10 AND ADD CONNECTIONS TO U80 AS SHOWN.
9. CUT CONNECTION BETWEEN U81 PIN 1 AND 16. GROUND PIN 16 AND INSURE CONNECTION BETWEEN PIN 1. AND DS6
8. CUT CONNECTIONS BETWEEN U81 PIN 2 AND U80 PIN 8 AND PIN 3 AND J3 PIN 38 AND CONNECT NEW CIRCUITRY AS SHOWN.
7. CUT PC RUN BETWEEN E18 AND J2 PIN 37. CONNECT NEW CIRCUITRY AS SHOWN.
6. CONNECT TO PARALLEL INTERFACE +5 VOLT BUS. USED TO SUPPLY +5 VOLTS TO TEST FIXTURE.
5. CUT ORIGINAL CIRCUITRY FROM E43 AND U81 PIN 14 AND U81 PIN 13 AND CONNECT NEW CIRCUITRY AS SHOWN.
4. CUT ORGINAL CIRCUITRY FROM E39 AND CONNECT NEW CIRCUITRY AS SHOWN.
3. CUT ORIGINAL CIRCUITRY FROM E17,19,21 AND CONNECT NEW CIRCUITRY AS SHOWN
CONNECT SPARE PINS 45 AND 46 AS SHOWN.
2. LEAVE CIRCUIT FROM J3 PIN 39 TO E50 INTACT. CUT REMAINDER OF CIRCUITRY BEYOND E50.
CONNECT NEW CIRCUIT TO E50 AS SHOWN.
1. // SYMBOL IS USED TO INDICATE NEW CUT POINT.
—•—•— IS USED TO INDICATE NEW WIRING CONNECTION.

NOTES



ORIGINAL PAGE IS
OF POOR QUALITY

FOLDOUT FRAME 2

Figure 4.2-1: Graphic 7 Parallel Interface Schematic for Harris Host

4.2.2.6 READ/WRITE Control ($\overline{R}/W$). The $\overline{R}/W$ control signal was added to the Graphic 7 parallel interface to allow the Harris communications handler to determine if the last I/O operation which caused an IIFU signal was an input or an output. Existing digital logic hardware on the Sanders parallel interface card was rewired to provide this signal. The modifications are detailed in Figure 4.2-1. Flip-flop (FF) U80B was defined as the READ/WRITE FF for the parallel interface. It was determined that the Graphic 7 INIT signal would not be used in the CAPS application. For this reason, output line J2-47 was redefined as the $\overline{R}/W$ signal line. Inverter U82F was used to obtain the correct polarity signal to provide an IIFU signal whenever the data on the OD bus was read and to set the $\overline{R}/W$ FF. Driver U81F was used to provide the necessary signal level at output J2-47. The output of the IWR NAND gate U78C is used to clear the $\overline{R}/W$ FF. In this manner, it is possible to obtain a direct indication of the cause of the IIFU signal.

4.2.2.7 Input Interrupt From Unit (IIFU). In the standard Graphic 7/Harris parallel interface this signal is used to interrupt the Harris central processor whenever the Graphic 7 has data for input to the host. A modification to this process was added to allow for an interrupt of the Harris whenever output data was taken by the Graphic 7. This is necessary because of the manner in which teletype data is transmitted from the Harris UBC by the Harris parallel software handler. Figure 4.2-1 again details both the standard Harris interface version of the signal's digital logic and the new TTY handler modification. On the standard Graphic 7/Harris parallel interface board, NAND gate U78D is used as an inverter for the IWR input signal to set the proper output polarity on the IIFU signal. The CAPS modification shows one input of this NAND gate being driven by the $\overline{IWR}$ signal while the other input is driven by the $\overline{OWR}$ signal. In this manner, whenever either IWR or OWR is sent, an IIFU signal is also sent.

4.2.2.8 Connect (CNCT). The connect signal was added to the standard Harris/Graphic 7 parallel interface board to allow operation with the Harris ABC/CBC/UBC series of parallel controllers. The daisy-chain nature of these controllers requires that each peripheral unit on the chain have the capability to monitor the Unit Code Register (UR00-03) to determine who should respond to any given I/O operation. The CNCT

signal is the indication that the last device addressed by the Harris controller is still listening. It is also used as the enabling signal for each unit's output tri-state drivers (e.g. U81 and U66 in Figure 4.2-1). The Graphic 7 interface board is provided with a decoder section consisting of digital chips U73, U74 and U76. The outputs of these chips are used in conjunction with the DISC signal to set and clear the connect flip-flop U80A. The Q output of the CNCT FF is used to control the output state of the tri-state driver U81 as well as to provide the CNCT signal back to the Harris. In addition, the $\overline{Q}$ output is used to set the state of the tri-state drivers for the other output signals from the parallel interface board through inverters U82A and U82C. The fact that the $\overline{Q}$ output was also used to control the state of the driver for the CNCT signal created a problem of level detection for the BIU and the Harris UBC (The CNCT output driver returns to the high impedance state which appears as a +5V TTL level to the BIU/UBC before the change-of-state from high to low can be detected). Because of this difficulty the driver for the CNCT signal is forced to be on at all times to insure that the state change from high to low can be observed on the signal line whenever a DISC signal is sent. Figure 4.2-1 indicates this modification by having U81-15 (by U81D) attached to ground and the CNCT signal going through driver U81E. Note that LED DS6 is the connect indicator on the parallel board and is on whenever the $\overline{Q}$ output is low.

4.2.2.9 Disconnect (DISC). As mentioned above, this signal is used by the Harris UBC to force all units attached to a particular controller to drop the CNCT signal and inspect the UR bus to determine which unit is being addressed by the next I/O operation. On the Graphic 7 parallel interface board, the DISC signal resets the CNCT FF and enables the output of the decoder section through NAND gate U78B. This enabling signal is held on, however, by the one-shot U79A. The output of the decoder section must be read within 50 nanoseconds (ns) of the DISC signal falling or the one-shot disables the output of the decoder section and prohibits the setting of the connect FF.

4.2.2.10 Unit Code Register (UR00-03). The Harris chaining controllers can be connected via a daisy-chain to as many as 16 devices at one time. This requires the use of a four-bit unit code to indicate the referenced device. The input from the UR bus is used to drive the decoder section of the parallel interface card. The set of eight switches pro-

vided in U76 allows for the selection of either inverted or non-inverted input and the setting of the address to which the interface board responds. In the CAPS application, switches 2, 4, 6, and 8 are closed (ON) to select address 0 for the Graphic 7.

SECTION V

CAPS BIU HARDWARE

5.0 INTRODUCTION

Hardware used in the CAPS BIU for both the Graphic 7 display system and its Harris host computer is adapted from existing BIU designs used for the NASA Trend Monitoring System (TMS) and other similar systems. The main design feature which distinguishes the CAPS BIU is the device data interface. In most parallel interface designs, a bidirectional data bus is used to transfer 16 bits of parallel information. The design of the Harris parallel interface requires that 32 unidirectional lines be used for this purpose. In addition to the different data line requirement, several of the handshaking lines are unique to the CAPS application. To insure proper response by the connected device, additional control lines are added to allow for peripheral unit identification.

5.1 Basic Hardware

The digital logic of the CAPS BIU is mounted on a single board measuring approximately 7 x 7 inches. It is connected to the BIU chassis, modems and device interface connector by means of a 122-pin edge connector. The chip layout for the CAPS BIU is depicted in Figure 5.1-1. It revolves around the Mos Technology MCS6502A central processor chip. The hardware used in the basic BIU to interface with the cable communications system will not be discussed here, however, the device interfaces to both the Harris UBC and the Sanders Graphic 7 are explained in detail. For further information on the cable interface the reader is directed to [10] and [18]. Figure 5.1-2 is a schematic diagram of the CAPS BIU digital logic board.

5.1.1 Major Components

The BIU is composed of a microprocessor-based digital logic board, an RF modulator, an RF demodulator and a solid state power supply. All of these components are housed in a metal box which measures approximately 10 x 12.5 x 3 inches and weighs about 9 pounds. The unit consumes about 10 watts

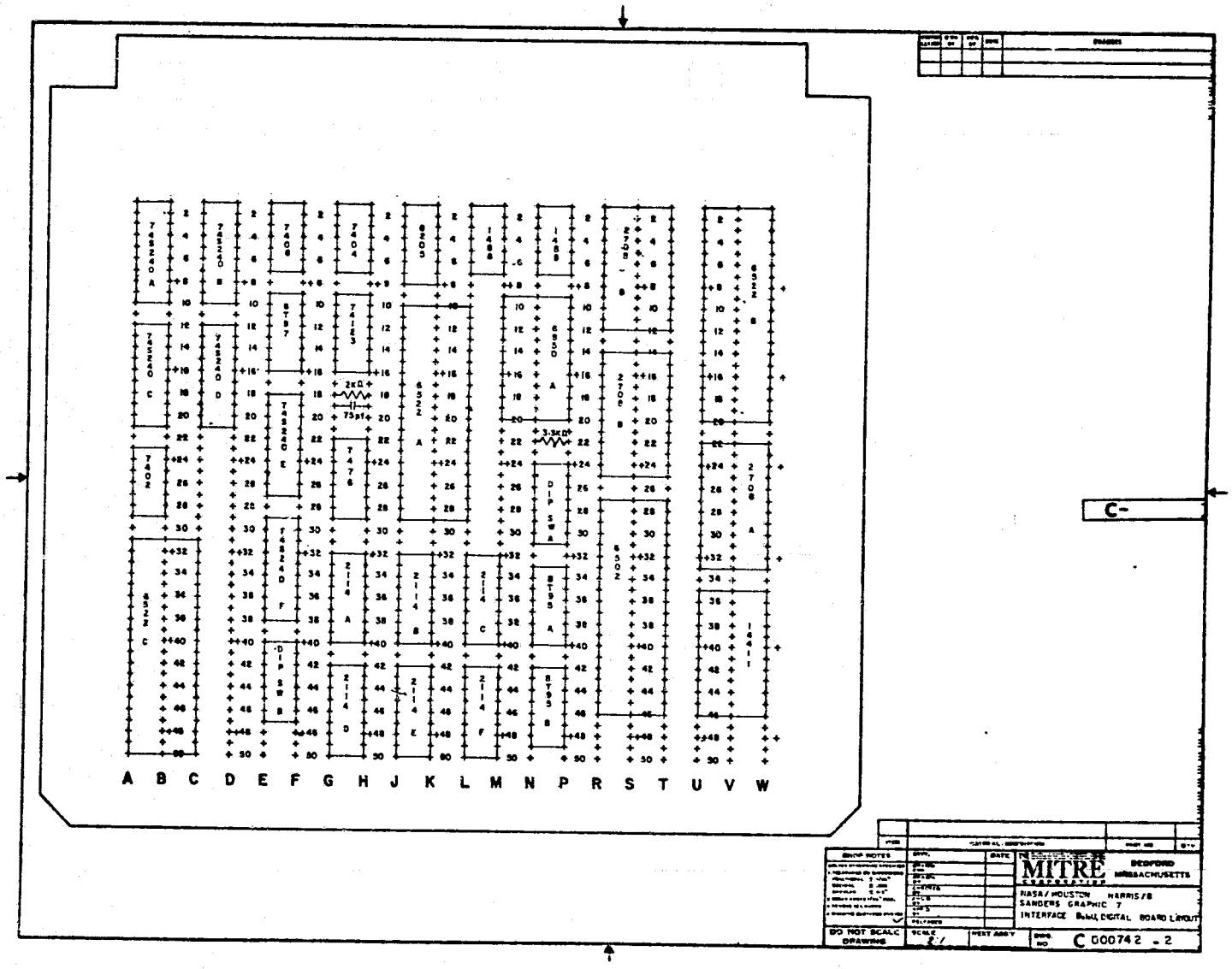
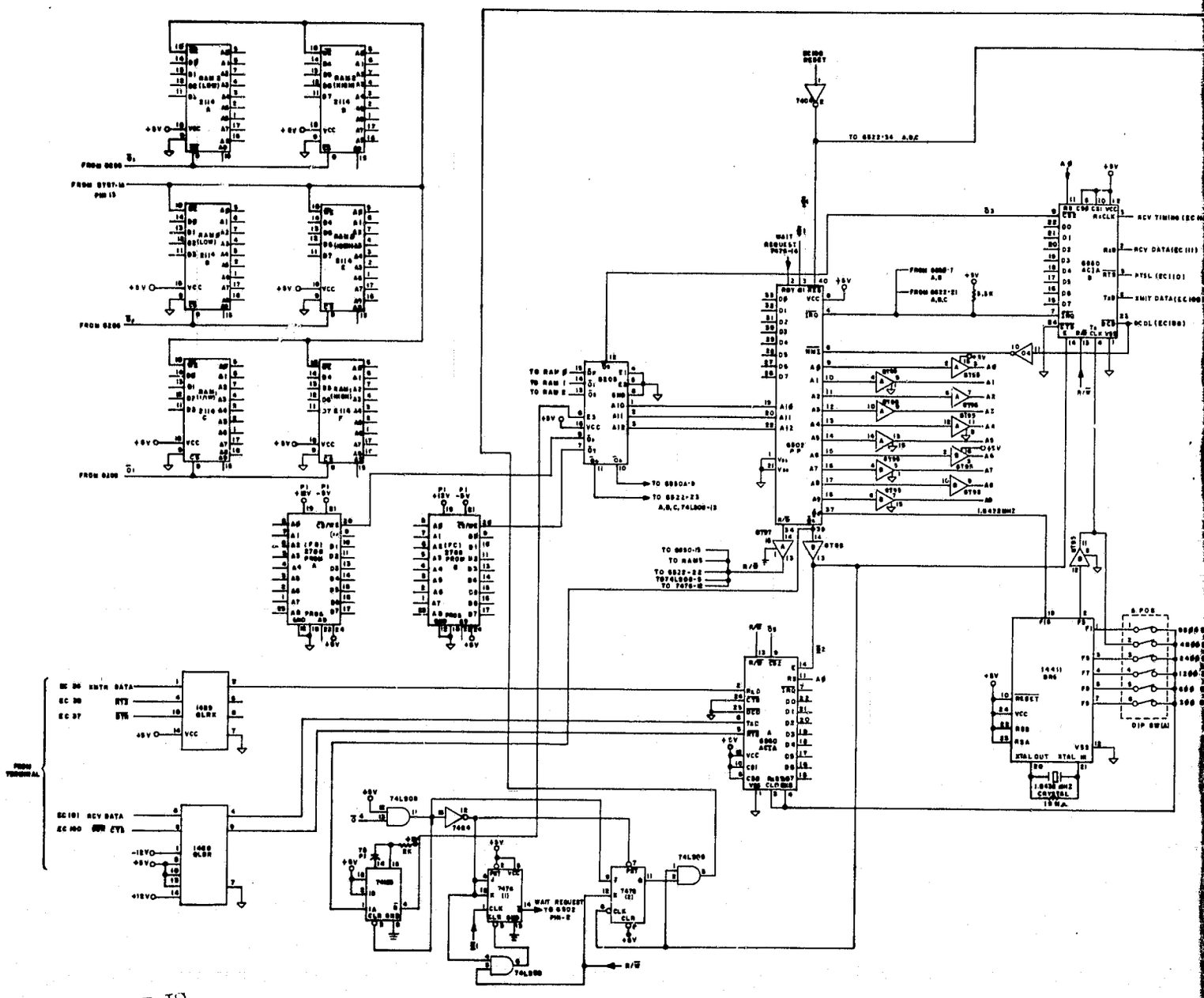
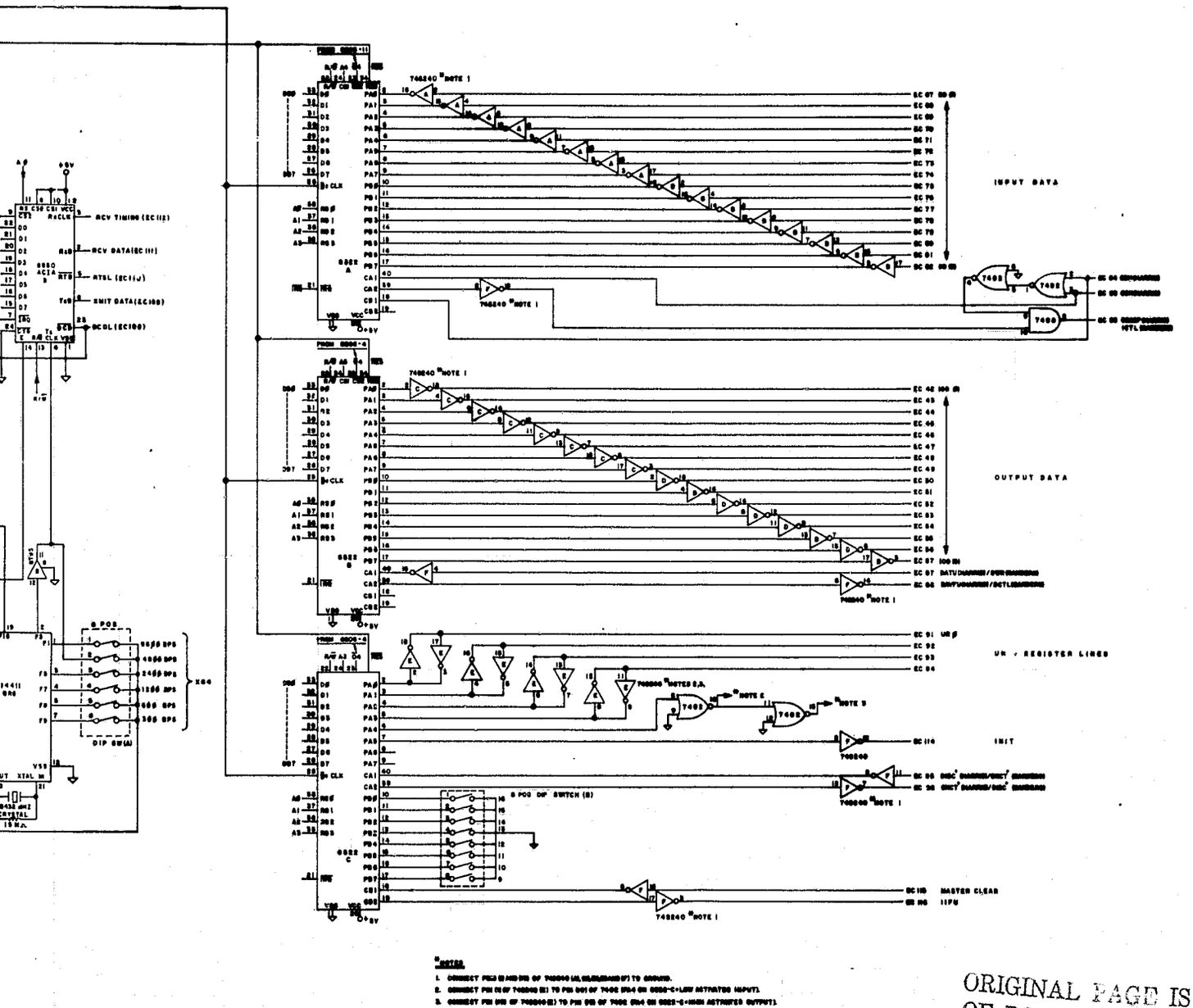


Figure 5.1-1: Chip Layout of the CAPS BIU Digital Logic Board



ORIGINAL PAGE IS
OF POOR QUALITY

SOLDOUT FRAME



ORIGINAL PAGE IS
OF POOR QUALITY

FOLDOUT FRAME

Figure 5.1-2: Schematic Diagram of the CAPS BIU Digital Logic Board

of power. Further information on the RF modulator and demodulator may be obtained from [18]. The following paragraphs discuss the digital logic which is contained on the digital logic board.

5.1.1.1 Central Processing Unit (CPU). The NMOS MCS6502A microprocessor is used as the main logic element of the BIU. It was chosen for the design because of its high speed and powerful instruction set. Another consideration was that its idiosyncrasies were well understood by the designers.

The 6502A is an 8-bit CPU with two 8-bit index registers, an 8-bit accumulator and a software-selectable, variable length (256-byte maximum) hardware-managed stack. The chip supports direct, indirect and indexed modes of address. The CAPS BIU operates with a 1.8432 megahertz (MHz) crystal-controlled clock which allows an instruction cycle of approximately 550 ns.

5.1.1.2 Memory. Two 1024 x 8-bit ultraviolet-erasable programmable read only memory (EPROM) chips (Intel 2708 or equivalent) are used to contain the interface software which defines the specific application of the BIU. This software is described in Sections VI, VII and VIII of this report. The nominal access time for these chips is 450 ns.

Six 1024 x 4 bit NMOS static random access memories (RAM) are used to provide storage space for the software defined variables, the hardware stack and the packet input and output buffers. These chips provide sufficient space for 22 buffers of 128 bytes each. A memory map of the CAPS implementation is shown in Figure 5.1-3 below.

5.1.1.3 Parallel Interface Chips. Three Mos Technology MCS6522 Versatile Interface Adapter (VIA) chips are used to provide the necessary parallel data channel interface to the devices. One chip serves as the input interface, one as the output interface and the third provides an interface for various handshaking and control signals. Each chip has the capability of providing 16 input or output lines under software control. Eight bits of the third VIA interface are dedicated to providing a switch-selectable address input. With the

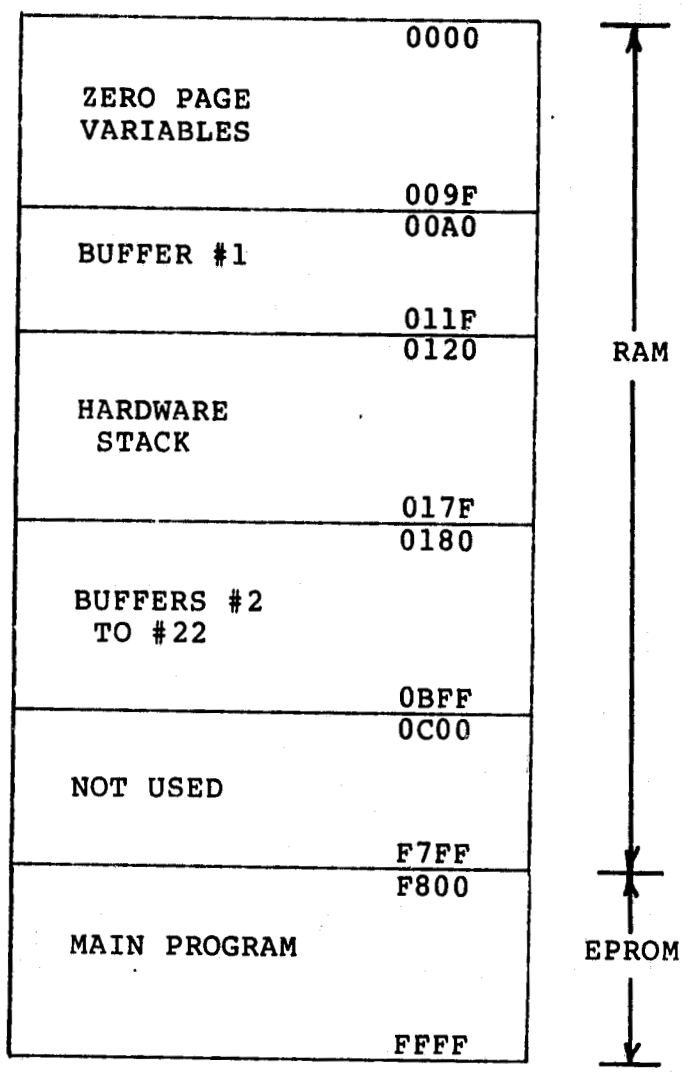


Figure 5.1-3: Memory Address Map of the CAPS BIU

addition of this 8-bit interface to the VIA, the low-order 8 bits of the 16-bit address word can be set through an 8-position DIP switch (See Paragraph 5.2.6).

As an added feature of the VIA, a clock mechanism is provided to allow the software to execute certain routines based on the system time. Only one of the six possible timers on the digital board is utilized by the current application. This timer provides a 10 millisecond (msec) clock for status message and sign-on response timing. For more information on the 6522 VIA chip the reader is directed to [19].

5.1.1.4 Serial Interface Chips. Commonality among the existing BIUs at NASA was maintained by the inclusion of the Motorala 6850 asynchronous communications interface adapter (ACIA) chips. While the device ACIA is contained on all digital boards, no facility is provided to allow use of this serial interface on the CAPS BIU chassis due to the absence of a serial interface connector. This feature, however, allows the CAPS BIU digital board to be removed from the CAPS chassis and inserted into a standard serial BIU chassis to function as a serial BIU. Alternatively, a CAPS BIU chassis could be altered in accordance with Table 3.2.2-II in [10]. Problems involving common components shared by the parallel and serial ports can be isolated by using the two types of chassis during checkout of the CAPS BIU digital board. In addition, the same type of chip is used to interface with the communications cable through the modulator/demodulator (MODEM) pair. The network ACIA chip is used to convert the 8-bit parallel data presented on the microprocessor data bus into a serial bit stream for use by the MODEM. The functioning of the 6850 chip is discussed in more detail ir [10] and [20].

5.1.2 Address Bus

The MCS6502A with its 16-bit address bus and 8-bit data bus has the capability of addressing 65536 bytes of memory or a lesser amount of memory and numerous peripheral devices. In the CAPS BIU, thirteen bits of the address bus are used to address the memory detailed in Figure 5.1-3 and the peripheral devices discussed above. By using address bits A10 through A12 to drive the 8205 decoder, chip select signals are provided to the various memory chips and the parallel and serial I/O devices. Table V is a list of the digital chips and the addresses used by the CAPS BIU to accomplish chip

selection. Each output of the address bus from the 6502 chip is an input to an 8T95 driver chip to insure the signal level at each device on the bus is a TTL level.

TABLE V

CHIP SELECTION ADDRESS SCHEME

DEVICE	ADDRESS (HEX)	SELECTION BITS A4	A5	A6	A10	A11	A12
RAM0	0000-03FF	0	0	0	0	0	0
RAM1	0400-07FF	0	0	0	1	0	0
RAM2	0800-0BFF	0	0	0	0	1	0
ACIA(B)	0C00-0C01	0	0	0	1	1	0
VIA0	1010-101F	1	0	0	0	0	1
VIA1	1020-102F	0	1	0	0	0	1
VIA2	1040-104F	0	0	1	0	0	1
ACIA(A)	1400-1401	0	0	0	1	0	1
EPROM0	F800-FBFF	0	0	0	0	1	1
EPROM1	FC00-FFFF	0	0	0	1	1	1

5.1.3 Data Bus

The 6502 uses an 8-bit data bus to communicate with all peripheral devices. It can be seen in the schematic diagram in Figure 5.1-2, that all the 6522 VIA's and the 6850 ACIA's as well as the 2708 EPROM have an 8-bit interface with the data bus. The RAMs, however, are configured with each chip interfacing to only 4 bits of the data bus thus requiring two chips to make up each 1024 bytes of memory.

5.2 Parallel Interface Hardware

As was mentioned above, the BIU uses the MOS Technology MCS6522 VIA as the parallel interface chip for all devices. In the CAPS BIU, two VIA's are used to allow 16 bits of input and 16 bits of output. The following paragraphs describe the

hardware implementation of the parallel interface to both the Sanders Graphic 7 and the Harris UBC. Table VI provides a quick reference to the chip location on the digital board.

5.2.1 Input Data

The 16 bits of parallel input data for the BIU enter through the 50-pin parallel chassis interface connector on pins 2 through 17 and proceed directly to digital board edge connector pins EC67 through EC82 (refer to Table III above). Each data bit is buffered by the two 8-bit 74S240 inverter/-driver chips (A) and (B) shown in the schematic Figure 5.1-1. The outputs of these buffer chips are then connected to VIA chip 6522(A) at digital board coordinate J-10 (see Table VI). The low order 8 bits are connected to the PA input and the high order 8 bits are connected to the PB input. The control lines for the 74S240 tri-state chips are connected to ground to insure that data is always fed through on the input bus.

5.2.2 Input Control

To allow the proper interface handshaking procedure to be followed, the control lines to and from the 6522 chip must be processed through the AND/NOR circuit shown in the schematic Figure 5.1-2. This circuit is designed to cut off the return handshake level when the attached device drops its valid data (e.g. IWR, ODH, etc.) signal level. As can be seen in the schematic, two possible handshake signals can be processed by the BIU. The one which enters on pin 19 (ODH/-IWR) of the parallel connector and is passed to EC83 is used to activate the CA1 interrupt of the 6522. The second signal, which enters on pin 20 (CDH) and is passed to EC84, is used to activate the CB1 interrupt. Both signals are passed directly to the VIA and are positive logic levels (i.e. active high, +5V TTL). In addition to activating the interrupts of the 6522, both signals are used to allow the output of CA2 (ODACP/ICTL) through inverter/driver 74S240(F) and the 7408 AND gate. This level is then seen by the attached device as a positive handshake signal which can only be active while the device holds its input valid signal high. There is a delay equivalent to three TTL gate delays (approximately 36 nsec) between the dropping of the input valid signal and the falling of the return handshake signal. The CB2 output of this 6522 is not used in the current application. The tri-state control for the 74S240(F) chip, used to drive the con-

TABLE VI

BIU CHIP LAYOUT DIGITAL BOARD CROSS REFERENCE

CHIP NUMBER	FUNCTION	DIGITAL BOARD COORDINATES (PIN #1)
1488	SERIAL LINE DRIVER	L-1
1489	SERIAL LINE RECEIVER	N-1
14411	CLOCK	U-35
2114(A)	RAM (0000-03FF, Bits 0-3)	G-32
2114(B)	RAM (0400-07FF, Bits 0-3)	J-32
2114(C)	RAM (0800-0BFF, Bits 0-3)	L-32
2114(D)	RAM (0000-03FF, Bits 4-7)	G-42
2114(E)	RAM (0400-07FF, Bits 4-7)	J-42
2114(F)	RAM (0800-0BFF, Bits 4-7)	L-42
2708(A)	EPROM (F800-FBFF)	U-22
2708(B)	EPROM (FC00-FFFF)	R-14
6502	CPU	R-27
6522(A)	VIA PARALLEL PORT (INPUT)	J-10
6522(B)	VIA PARALLEL PORT (OUTPUT)	U-1
6522(C)	VIA PARALLEL PORT (CONTROL)	A-31
6850(A)	SERIAL DEVICE UART	M-9
6850(B)	NETWORK UART	R-1
7402	QUAD NOR GATE	A-23
7404	INVERTER (6-BIT)	G-1
7408	QUAD AND GATE	E-1
7476	J-K FLIP-FLOP	G-22
74123	ONE SHOT	G-9
74S240(A)	8-BIT INVERTER/DRIVER (INPUT DATA)	A-1
74S240(B)	8-BIT INVERTER/DRIVER (INPUT DATA)	C-1
74S240(C)	8-BIT INVERTER/DRIVER (OUTPUT DATA)	A-12
74S240(D)	8-BIT INVERTER/DRIVER (OUTPUT DATA)	C-12
74S240(E)	8-BIT INVERTER/DRIVER (UNIT REG.)	E-18
74S240(F)	8-BIT INVERTER/DRIVER (CONTROL LINES)	E-29
8205	ADDRESS DECODER	J-1
8T95(A)	6-BIT INVERTER/DRIVER (ADDRESS 0-5)	N-33
8T95(B)	6-BIT INVERTER/DRIVER (ADDRESS 6-9)	N-42
8T97	DRIVER	E-9
DIPSW(A)	SERIAL DEVICE SPEED SELECTION	N-24
DIPSW(B)	BIU ADDRESS SELECTION	E-40

NOTE: When more than one chip of a given type is used, each chip is given a suffix letter (x) to allow specific definition.

trol signals for the input VIA as well as the the other two VIAs, is also held at the zero TTL level to insure all signals are valid at all times.

5.2.3 Output Data

The output data lines originate on the PA and PB outputs from 6522(B) as shown in the BIU schematic. As with the input bus, the output lines are buffered by a pair of 74S240 inverter/drivers (C) and (D) whose control lines are held at ground potential (0V TTL). This insures that the data lines can be read at any time by the connected device. The low-order 8 bits of output are on the PA bus and the high-order 8 bits are on the PB bus. The outputs of the two 74S240 chips are connected directly to edge connector pins EC42 through EC57. From the edge connector, the output data lines proceed directly to the 50-pin chassis connector where they terminate on pins 22 through 37.

5.2.4 Output Control

As in the case of the input bus, the BIU provides a fully handshaken level control of the output data flow. When data is written into the output VIA, the signal on the CA2 output transitions from a high state (+5V) to a low state (0V). The connected device sees this as a low-to-high transition through the inverter/driver chip 74S240(F). This signal (DAVFU/OCTL) is present on 50-pin connector pin 40 which is connected to EC88. The connected device interprets this signal as indicating that valid data is present on the output data bus. The signal could be used to trigger an input interrupt of the device; however, in the CAPS BIU, a separate signal line is used for the input interrupt. The connected device signals (DATU/OWR) to the BIU that it has accepted the data on the output bus by setting a +5V TTL level on the return handshake line which enters on pin 39 of the 50-pin connector and proceeds directly to another inverter/-driver through EC87. The output of this driver is connected to the CA1 input on the VIA. A high-to-low transition on CA1 resets CA2 which in turn causes the connected device to reset the CA1 input. The VIA is now ready to process another output data word. Neither the CB1 nor the CB2 signals are used on this VIA in the current CAPS application.

5.2.5 Other Handshake and Control Signals

The two VIAs mentioned in the paragraphs above are the primary devices used to transfer data to and from the BIU. The third VIA on the digital board, 6522(C), is used to enable the BIU to duplicate the Harris UBC parallel interface by handling the remainder of the interface signals. These signals include the connect/disconnect, unit identification, interrupt, and READ/WRITE signal.

5.2.5.1 Connect/Disconnect Signals. The Harris UBC employs a connection handshake sequence using the CNCT and DISC signals to allow the operation of the UBC on a daisy-chained I/O bus. Since the BIU can be acting as either the Harris UBC or the Graphic 7 display, the CNCT and DISC signals reverse function depending on the connected device. These signals are generated by the software executing in the BIU and thus the signal line definition is under software control. When the BIU is connected to the Harris UBC the signal entering on pin 45 of the 50-pin connector and proceeding through EC95 and the 74S240(F) inverter/driver to CA1 of 6522(C) is interpreted as the DISC signal. If the Sanders Graphic 7 is connected to the BIU, the same signal line is defined as the CNCT signal. In a like manner, the signal leaving the BIU from CA2 via EC96 and pin 46 of the 50-pin connector is assumed to be the CNCT signal when the BIU is attached to the UBC and the DISC signal when attached to the Graphic 7. Control of the outputs from CA2 and interpretation of the input on CA1 is discussed in the appropriate software discription for the two possible cases (See Paragraphs 8.5 and 9.1.2).

5.2.5.2 Unit Register Bus. As in the case of the CNCT/DISC signal lines, the UR bus lines can be interpreted in two ways depending on the device to which the BIU is connected. In the case of the Harris UBC, the signals originate in the UBC and must be received by the BIU as input and PA0 through PA3 of 6522(C) must be set to read the lines. When the BIU is connected to the Graphic 7, PA0 through PA3 must be set to write on the lines since the unit address is generated in the BIU. For this reason, the dual inverter/driver arrangement shown in the BIU schematic is used. The output of PA4 on 6522(C) is used to select the appropriate half of 74S240(E) which allows the UR bus to be either an input to or an output from the BIU. The control of these signal lines is also discussed in more detail in Paragraphs 8.5 and 9.1.2.

5.2.5.3 Input Interrupt. Whenever the Graphic 7 has input data for the Harris or whenever an output data word has been processed, the Graphic 7 must provide an interrupt on the IIFU signal line. There is no similar signal which is returned to the Graphic 7 by the UBC so this can be considered a one-way control line, that is, it is only active when the BIU is attached to the UBC. The BIU uses the output from CB2 to signal the Harris UBC in the same manner as the Graphic 7 signals with the IIFU line. The software description of this signal explains how the signal is generated (see Paragraphs 8.2.3.2 and 8.3.2.2). Since no true hardware handshaking takes place in the BIU, the signal is totally under software control. It is inverted by the hardware in the BIU and then fed directly to EC116 and pin 49 of the 50-pin connector.

5.2.5.4 $\overline{R}/W$ Control. To allow the Harris UBC to determine which of the two conditions (input/output IIFU) mentioned above exists when the IIFU signal is triggered, the $\overline{R}/W$ signal line is used. As in the case of the IIFU signal, this is a one-way signal from the Graphic 7 to the UBC. There is no counterpart going to the Graphic 7 from the UBC. For this reason this signal is only active when the BIU is connected to the UBC. Again this signal line is under software control and is documented more fully during the software discussion (Paragraphs 8.2.3.1 and 8.3.2.1). As can be seen from the BIU schematic, the signal proceeds directly from the PA5 output through inverter/driver 74S240(F) to EC114 and then to pin 47 of the 50-pin chassis connecter.

5.2.6 Manual Address Selection

The CAPS BIUs contain a DIP switch which allows the user to select the bus address to which a BIU will respond. The 8-bit DIP switch is located at the input to the PB bus of 6522(C) at digital board reference E-40. By selectively opening and closing these 8 switches, the user can set a one-byte address to the VIA. The BIU software uses this address whenever it must compare a packet address to its "home" address. Like several of the above listed features, this function is under software control and is discussed in more detail in the appropriate paragraph (Paragraph 7.11.3.2). The hardware configuration, however, indicates that when a given switch is open, there is a TTL one (+5V) present in that bit

location. Likewise, when a given switch is closed, there is a TTL zero (0V) present. The high-order bit is at digital board reference E-47 and the low-order bit at reference E-40.

5.3 Serial Interface to Network

The communications interface to the bus communications system is through the Motorola 6850 ACIA chip at digital board coordinate R-1. The chip provides a TTL interface to the BIU MODEM modules for establishing the proper signal timing and format used on the coaxial cable system. A detailed discussion of the functions of these components and the other hardware related to the transmission and reception of data to and from the cable system can be found in [10]. A discussion of the details of the MODEM design can be found in [18].

5.4 Serial Terminal Interface

As was mentioned above, the hardware necessary to allow the digital board to function in a serial terminal BIU is installed on all CAPS BIU digital boards. No detailed discussion of this hardware will be undertaken here. It is sufficient at this point to define the chips on the digital board which are dedicated to this function. The 6850 chip at coordinate M-9 is the ACIA chip used to communicate to an RS-232C interface. It uses the 1488 line driver chip and the 1489 line receiver chip to insure proper signaling. DIP switch (A), at coordinate N-24, is used to select the line speed for the serial interface through the 14411 clock chip. A more detailed description of the functioning of this type of interface can be found in [10].

PART III

CAPS COMMUNICATIONS SYSTEM SOFTWARE

SECTION VI

STANDARD CAPS INTERFACE SOFTWARE

6.0 INTRODUCTION

This section discusses pertinent sections of the software used by the Harris host computer and the Graphic 7 controller to establish a parallel interface between these devices and also interface each device to the MITRE BIUs. A detailed discussion of the software used in both the Harris and the Graphic 7 can be found in [7] and [6] respectively. Only the elements directly affecting the operation of the BIU are addressed here. The block diagram appearing in Figure 2.1-2 represents the relationship of the software packages used to implement this interface.

6.1 Harris UBC Interface

The Harris UBC interfaces with the Graphic 7 as well as the BIU through a software communications handler which allows communications in either the teletype emulator mode, that is, word at a time through the "A" register, or the DMA mode, that is, directly from the memory locations indicated by a starting address and word count. The actual handler is a combination of the existing Harris TTY handler and the DMA portions of a handler in use by MIT Lincoln Labratories on a Harris system using a Chaining Block Controller (CBC). The CAPS handler was developed by NASA's Institutional Data Systems Division [7]. The standard VULCAN TTY handler does not support a parallel interface and thus must be merged with the parallel CBC interface handler which does not support TTY operations.

The Harris UBC parallel interface is designed to transmit 24-bit words to peripheral devices via a daisy-chained interface. In the CAPS application, the Sanders Graphic 7 uses only the low-order 16 bits of the interface for parallel communication. Standard Harris I/O service commands [21] are used to perform the transfer operations with the exception of the following changes:

1. Paper tape and cassette functions have been removed.

2. Special purpose keys CTL-E, CTL-R, and CTL-L are not implemented.

3. Function code '04 is invalid.

4. Function code '07 is used for DMA read.

5. Function code '10 is used for DMA write.

(Note: The notation {'} indicates an octal number.)

6.1.1 Teletype Operations

Typical teletype data transfers are initiated by use of the symbolic I/O commands to output a data word (ODW) or input a data word (IDW). These transfers are used only when the Graphic 7 terminal is functioning as a TTY terminal. The input or output is always referenced to the Harris host. These data transfer commands function as described by the Harris I/O services manual [21]. Data words transferred in the TTY mode are composed of three bytes, the low-order byte containing valid data and the two high-order bytes containing null or zero. These operations are always word(byte)-at-a-time through the "A" register with the Slash 8 processor maintaining the application software in a wait condition until the transfer is acknowledged. The completion of each word transfer is signaled by an interrupt from the device allowing the I/O handler to process the next word.

6.1.2 FSP Operations

Two types of transfer can occur when a FSP application program is running: parallel through the "A" register, and DMA. The commands described in Paragraph 6.1 and the I/O services manual [21] determine which type of transfer will be used.

6.1.2.1 Register I/O. If symbolic commands are used in the application program, the transfer of data occurs through the "A" register. As in the case of TTY transfers, each operation loads the data word into the "A" register via the appropriate command and causes the UBC to effect the trans-

fer. In the FSP mode, however, the two low-order bytes in each word contain valid data and special control characters are not recognized.

6.1.2.2 DMA Transfers. Since DMA operations, by definition, involve direct transfer from memory locations (not through any register), the parallel handler starts each DMA transfer with an output address word (OAW) command to the UBC. This sets the starting address for the following transfer. This address represents the first location where data will be stored on input, or the first location from which data will be read on output. The handler then executes and output command word (OCW) instruction which establishes the device involved with the transfer, the word count used and whether input or output will occur. Upon receipt of this command, the controller commences the transfer without interfering with the Slash 8 processor operations and interrupts the processor when the transfer is complete (word count zero interrupt). The UBC performs the DMA transfer by transmitting words to the communications interface hardware, two bytes at a time with valid data in both bytes, directly from the memory addresses specified without interfacing with the central processor. It is important to note that special TTY control characters can be embedded in these words and must be ignored by the communications media.

6.1.3 Message Buffering

To increase the speed of data transfer during certain CAPS application programs, a special communication package was developed which decreases the number of I/O operations for any given program using FSP. The package is transparent to the application and the communications media but may result in increasing the length of any given I/O operation. The main function of the package is to buffer output messages from the Harris FSP until critical commands are issued or the buffer becomes full. At that point a standard I/O operation is initiated and control returned to the application program. A double buffering scheme is used to prevent the application program becoming frozen in a wait state while the transfer is underway. More details on this message buffering package are available in [8].

6.2 Graphic 7 Interface

Like the Harris host, the Graphic 7 controller uses special software to interface with a parallel channel [6]. The standard Sanders Graphics Control Package (GCP) has the ability to detect when the parallel interface card is installed. It makes the assumption that if the card is installed, the host will communicate with the Graphic 7 via this parallel interface. The parallel or serial state of the interface is established by the parallel interface diagnostic routine which is executed whenever the Graphic 7 terminal is reset. If the diagnostic completes successfully, the IOINTF flag is set to 0 indicating communication via the parallel interface. If it fails, the flag is set to 1 and host communication is through the serial multiport board. In the case of parallel operations, the Graphic 7 utilizes a 16-bit bus for input and another 16-bit bus for output. Transfers are through the "R0" register in the TTY emulator mode and via register I/O and DMA in the system mode of operation. The transmission medium for the Graphic 7 parallel interface must be transparent to these operations.

6.2.1 Teletype Emulator

When initializing the Graphic 7, the user will normally place the Graphic 7 controller in the TTY mode by entering a carriage return when the test pattern appears on the screen and then typing "Y" and a carriage return. This initiates communication throught the TTY emulator section of GCP enabling the user to communicate with the host as if the Graphic 7 were a TTY device. In this mode, the 16-bit words on the buses are again divided into two bytes with valid data in the low-order location and null or zero in the high-order location. Three special characters are recognized by the emulator. These are carriage return (forces the cursor to the lefthand side of the screen), line feed (forces the cursor to move down the screen one line), and the graphic control character octal 235 (forces the controller into the system mode).

6.2.2 System Mode Input

As mentioned above, the system I/O handler performs two types of data transfers: parallel through the "R0" register and DMA. Here again the GCP determines the type of interface to the host by performing the parallel interface diagnostic. In this case the IOMODE flag reflects the result of the test.

The set and clear criterion used in the TTY emulator is used again. When communicating via the parallel interface, however, unlike the TTY emulator mode, the 16-bit data words contain two valid data bytes each.

6.2.2.1 Register I/O. Normal communication to the host in the system mode utilizes the "R0" register for transfer. When an interrupt from the host is detected in the parallel interface status variable (PARSTS), a one word transfer is assumed and the word placed in the "R0" register. This word is then compared to the valid message codes recognized by GSS-4 as host commands. These commands can result in further action by the GCP communication interface to set up for a DMA transfer, if necessary, or they can simply result in special processing functions necessary to the GSS-4 operation.

6.2.2.2 DMA Operation. If a DMA operation is indicated, two more words are read from the host; the first is the starting address where the data is to be stored and the second is the number of words in the transfer. This second word is also stored in "R5" and acts as a flag for DMA operations. Words processed by the handler when the DMA flag is set (greater than zero), are stored according to the addressing information transfered at the start of the DMA operation. The DMA operation continues until the word count has been satisfied and the word count zero bit in PARSTS is set. Unlike the Harris UBC DMA operation where the UBC controls the transfer while the CPU continues with other processing, the Graphic 7 GCP processor remains in a tight loop at this point until the operation is complete. Only those operations which generate processor interrupts continue during DMA transfers.

6.2.3 System Mode Output

In the system mode, all data transfers to the host are accomplished through register "R0". As in the case of input operations, both bytes of output words contain valid data. There are two types of output used in the system mode: a fixed length, 4-word message, and a variable length message. Each variable length message is preceded by a fixed length message instructing the HCP to prepare a buffer of the indicated length. Hardware interface handshaking is accomplished and controlled by the software storing into and then reading the PARSTS flag as an indication of the host's ac-

ceptance of the data. The GCP remains in a tight loop after setting the output handshake signal in PARSTS until it detects the return acknowledgment in PARSTS.

SECTION VII

GENERAL BIU SOFTWARE

7.0 INTRODUCTION

The software developed for the CAPS BIU was adapted from existing code operationally interfacing several serial and parallel devices to other cable sytems such as the TMS system at NASA [22]. Many of the subroutines used in these systems are common in whole or in part to subroutines in the CAPS system. This is markedly so in the case of the routines which interface with the bus communications cable. This section of the report documents these common sections of the code and portions of subroutines which function in a similar manner although using different variable names. Table VII is a list of the routines used in both the Harris UBC BIU software and the Graphic 7 BIU software. It indicates the degree of commonality and functions exercised by each routine. It can be used to obtain a more detailed understanding of the subroutine interactions when reading the the BIU software discussions in the following sections.

7.1 Reset Operation

When the power is first applied to the BIU or when the user presses the RESET button on the chassis, the BIU hardware forces the program counter to be loaded from the RESET hardware vector address $FFFC, where the "$" symbol indicates a hexadecimal value. In most BIU applications, this vector location is loaded with the starting address for the code. In all CAPS BIUs, the starting address used as the RESET vector address is $F800. The RESET code performs several initialization functions including initialization of the serial and parallel interface chips, zeroing most variables to establish known conditions, setting the pointers for the buffer address space and starting a Time-Of-Day (TOD) clock used by several software timers. The following paragraphs discuss each of these functions in some detail. The reader is directed to Appendices I and II to follow the discussion with flow charts and the source code. The assembler language used in the BIU software source code is documented in [23].

TABLE VII

BIU SOFTWARE SUBROUTINE INTERFACING

NAME	CALLED BY	CALLS	FUNCTION	* DEGREE OF COMMONALITY
RESET	Hdwr Vector INDEV	PCONST ENQ STIMER PUTSTR INTBUF (MATCH)	Initializes all variables and Peripheral Chips, Sets Up Buffers	1
MLOOP	MLOOP	NET OUTDEV INTBUF INDEV TIMOUT CKTOUT MLOOP	Main Program Loop to Continually Call all Main Subroutines	3
NET	MLOOP	SFINC DQ	Used to Transmit Packets on the Network	3
INDEV	MLOOP INDEV	PCONST STIMER PUTSTR ENQ INDEV RESET	Input Interface from Device	1
(SYSTMX)	RESET CKTOUT	----	Subroutine Section of INDEV Used during BIU Sign-On	1
(ASKAGN)	CKTOUT	----	Subroutine Section of INDEV Used during BIU Sign-On	1
OUTDEV	MLOOP	PCONST ENQ PUTSTR DQ STIMER	Output Interface to Device	1
TIMOUT	MLOOP	----	Increments the 1/4 Sec Clock	3

TABLE VII (Continued)

NAME	CALLED BY	CALLS	FUNCTION	* DEGREE OF COMMONALITY
CKTOUT	MLOOP	CTIME SNDSTA PUTSTR (ASKAGN)	Check for Various System Timeouts	2
CTIME	CKTOUT	----	Compares TOD Clock to Variable Timer	3
SNDSTA	CKTOUT	PCONST ENQ STIMER	Generates a Status Packet	3
INTBUF	RESET INDEV IRQ	ALLOC	Sets Up a Buffer for Input	3
IRQ	Hdwr Vector	SFINC ENQ INTBUF	Receives Packets from Network	2
NMI	Hdwr Vector	----	Resets the Network UART	3
PCONST	RESET INDEV OUTDEV SNDSTA	ALLOC	Initializes an Output Packet	3
STIMER	RESET INDEV SNDSTA OUTDEV	----	Sets a Variable Timer	3
SFINC	NET IRQ	----	Increment a Status Message	3
ALLOC	INTBUF PCONST	----	Reserves a Buffer for Input or Output	3

TABLE VII (Concluded)

NAME	CALLED BY	CALLS	FUNCTION	* DEGREE OF COMMONALITY
ENQ	RESET INDEV OUTDEV SNDSTA IRQ	----	Adds a Buffer to either Input or Output Queue	3
DQ	NET OUTDEV	----	Removes a Buffer from either Queue	3
PUTSTR	RESET INDEV OUTDEV CKTOUT	----	Sends BIU Dialogue to Device	1

* 1 - Differences in several consecutive lines of code in several parts of the routine.

2 - Differences in variable names and/or small areas of the code.

3 - Identical throughout.

NOTE: Names in parenthesis indicate entry points in INDEV Subroutine.

7.1.1 Chip Initialization

Two basic types of chips are initialized in this section of the code; the 6850 ACIAs and the 6522 VIAs. The initialization is accomplished by loading a data byte into each chip's control register(s).

7.1.1.1 6850 ACIA Initialization. To initialize the 6850 chips, a two-step procedure is used. First, the chip is cleared via the master reset command. This command is issued by storing any data byte with bits 0 and 1 set high (e.g. XXXXXX11) into the ACIA status register . This causes a known starting state to exist in the 6850 chip. The second step is to set up the operating parameters for the ACIA by loading a second command byte into the status register. In the case of the CAPS BIU software, the device ACIA is not used and the network ACIA is initialized with the data byte 11011000.

Bits 0 and 1 are used to set the counter divide clock to divide by one. This indicates that the ACIA will operate at the clock rate of 1/6 the CPU since the ACIA clock is obtained from the main system clock chip, the 14411, on the pin 2 output which is 1/6 the output of pin 19, the CPU clock. As mentioned above, the network ACIA transmits and receives at a rate of 307.2 Kbps.

Bits 2, 3 and 4 are used to select the data format used by the ACIA. The particular pattern of bits cited above selects a byte format consisting of 8 data bits plus an even parity bit plus one stop bit. Thus, each data byte transmitted on the network consists of 11 bits.

Bits 5 and 6 initialize the transmitter control. They establish the state which will be recognized as indicating the transmitter buffer empty and the Request-to-Send output. In the CAPS BIU the values selected are RTS high and the transmitting interrupt disabled.

Bit 7 is used to control the states that generate a receiver interrupt of the CPU on the IRQ input. In the CAPS BIU, with the bit set high, the receive data register being full, the receiver being overrun or a low-to-high transition of the Data Carrier Detect (DCD) signal results in an interrupt of the BIU CPU.

7.1.1.2 6522 VIA Initialization. Several different registers are initialized in each of the VIAs used in the CAPS BIU. These include the Data Direction Register (DDR), the Auxiliary Control Register (ACR), the Peripheral Control Register (PCR) and the Interrupt Enable Register (IER). Since the values loaded into these registers vary depending on the application of the BIU, the discussion of their function can be found in Sections VIII and IX. In general, however, it can be stated that the DDR is set to indicate which bits of the parallel interface are used for input and which for output, the ACR is used to select the software clocks desired, the PCR is used to set the chip handshaking control logic, and the IER is used to establish which handshake signals of the VIA will generate interrupts of the CPU.

7.1.2 Variable Initialization

The majority of the "Page Zero" variables used in the CAPS BIU programs are initialized to zero by a tight loop labeled ZLOOP in the source code. This loop uses the address of the variable XADDR as its stopping point to allow for a modification feature which permits the addition of page zero variables to the source software without the need to adjust absolute address values. Note that by adding more variables to the list of page zero variables before XADDR, the programmer is not required to adjust the start or stop address of the ZLOOP. XADDR is the first of the Page Zero variables which is not initialized. A small group of variables are initialized to -1 in a similar manner. These variables are used as an indication of the end of a queue or to establish a certain system state. This latter initialization is accomplished by the loop labeled FFLOOP.

7.1.3 Buffer Address Initialization

The control of the input and output queues is an important part of the BIU software. An excellent discussion of this control can be found in Section VIII of [22]. Note, however, that the CAPS BIU uses only the QIN and QOUT queues in a manner similar to the MODCOMP described in [22]. In addition, there are 22 buffers available in the CAPS application instead of the 21 cited in the TMS documentation. The RESET code initializes these control features by first establishing a software stack of unused buffers (BUFSTK) and then

setting up a list of starting addresses for the buffers themselves (HIPTR and LOPTR). Then a list of the buffer pointers is created in the order of their use (NEXT).

7.1.3.1 BUFSTK. Each 128-byte buffer in RAM is given a number according to the order in which it appears in memory. There are 22 buffers (numbered $00 to $15) in the CAPS BIUs. The tight loop, STACKI, is used to establish this numbering scheme.

7.1.3.2 Buffer Addresses. The first buffer is located in an area of unused memory between the hardware stack and the bottom of the page zero variables. The starting address of this buffer is set to $00C0 before the regular buffer address initialization loop is entered. The next 21 buffer addresses are then set by the loop BUFFRI in HIPTR and LOPTR starting at address $0180 and continuing to address $0B80.

7.1.4 Start of TOD Clock

Each BIU periodically transmits a packet indicating its status. The CAPS BIU sends this packet approximately once every minute. To start the clock which keeps track of this process and all of the other processes that are based on the RESET time, the routine initializes the 6522 clock on PORT 1 to set a bit in the port's Interrupt Flag Register whenever approximately 1/100 second has elapsed. This is accomplished by loading $4800 into the PORT 1 timer and setting the value 01000000 in the ACR to indicate a free running clock. This means the clock will automatically restart whenever the clock decrements to zero and the interrupt flag will be cleared whenever the value of the low-order timer byte is examined (See TIMOUT Subroutine Paragraph 7.6).

7.1.5 Sign-Off Message

The CONECT flag is not reset during either the ZLOOP or the FFLOOP, therefore, it is an accurate indication of the previous state of the BIU prior to the RESET routine being entered. If the reset sequence is a result of the power-up cycle, then the data in CONECT will be some arbitrary value but most likely not 1. The routine checks the value of CONECT and if found to be 1, which indicates that the BIU was previously connected to another BIU, it generates a Sign-Off

message to the BIU address found in XADDR, which is also uncleared as a result of the two initialization loops. The purpose of this Sign-Off message is to insure that a BIU does not remain in a connected state and thus not free to communicate to other units when a previously connected BIU is reset.

7.2 MAIN LOOP

The RESET routine transfers control of the program to the MAIN LOOP. This loop repeatedly calls each of the main subroutines on a equal basis. The subroutines called and their functions are listed in the table below. Input from the network is accomplished via an interrupt (serviced in the IRQ routine) whenever the receive data register of the network UART becomes full.

TABLE VIII

MAIN SUBROUTINES

NAME	FUNCTION
NET	Transmits data packets on the network from QOUT
OUTDEV	Sends data into the device from QIN
INTBUF	Insures there is a data buffer available for IRQ
INDEV	Collects data from the device, packetizes it and adds it to QOUT
TIMOUT	Polls the 1/4 second timer
CKTOUT	Checks critical timer controlled functions

7.3 NET Subroutine

The main output software interface with the cable network is through the NET subroutine; therefore, this routine should be common to all subscribers on the network. Different applications may vary the routine to take advantage of different delays necessary to clear buffers or wait for acknowledgments (ACKs), but the main functions remain the same. This routine manages the transmission of data onto the network by controlling; collision avoidance, packet flow, LWT, data transmission, ACK recognition and output buffer freeing. Figure 7.3-1 is a flow chart of this routine.

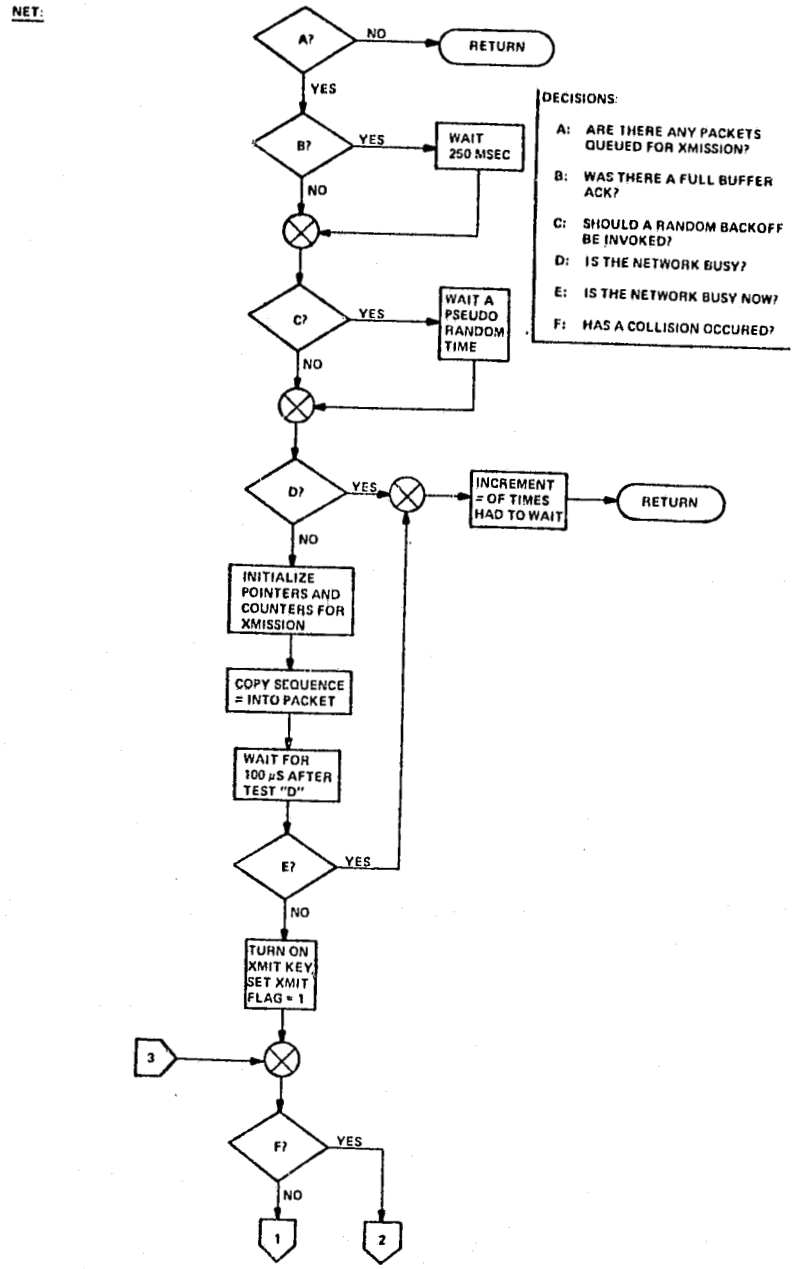
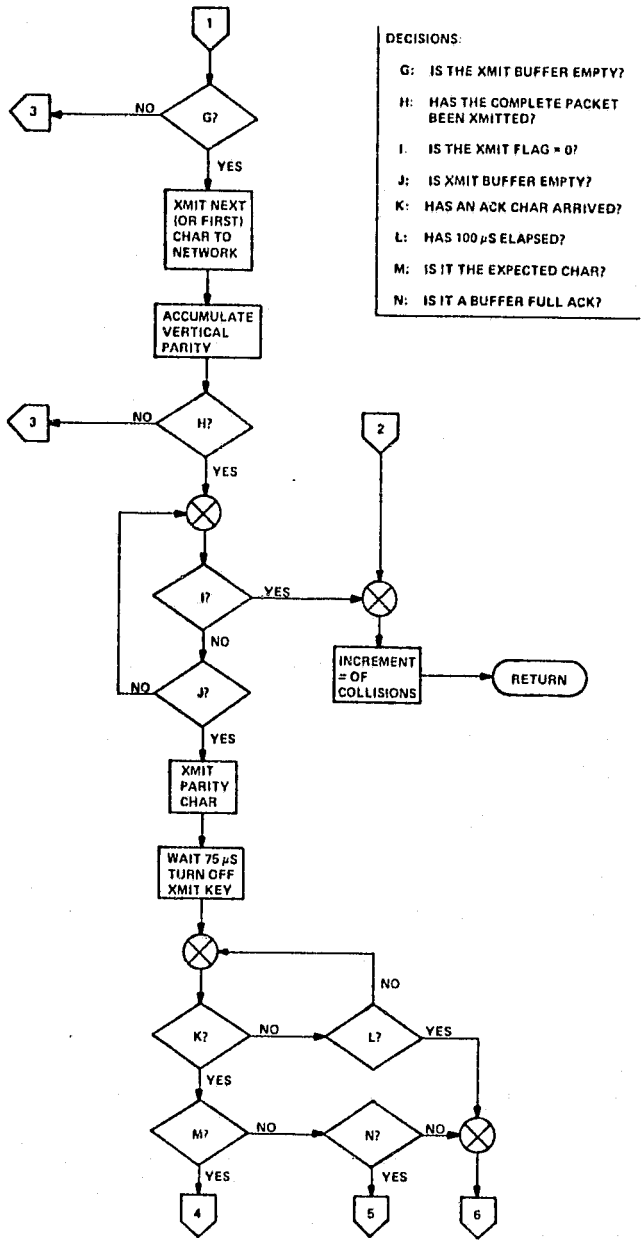


Figure 7.3-1: NET Subroutine Flow Chart

NET: (CONTINUED)



IC-56853

Figure 7.3-1: (Continued)

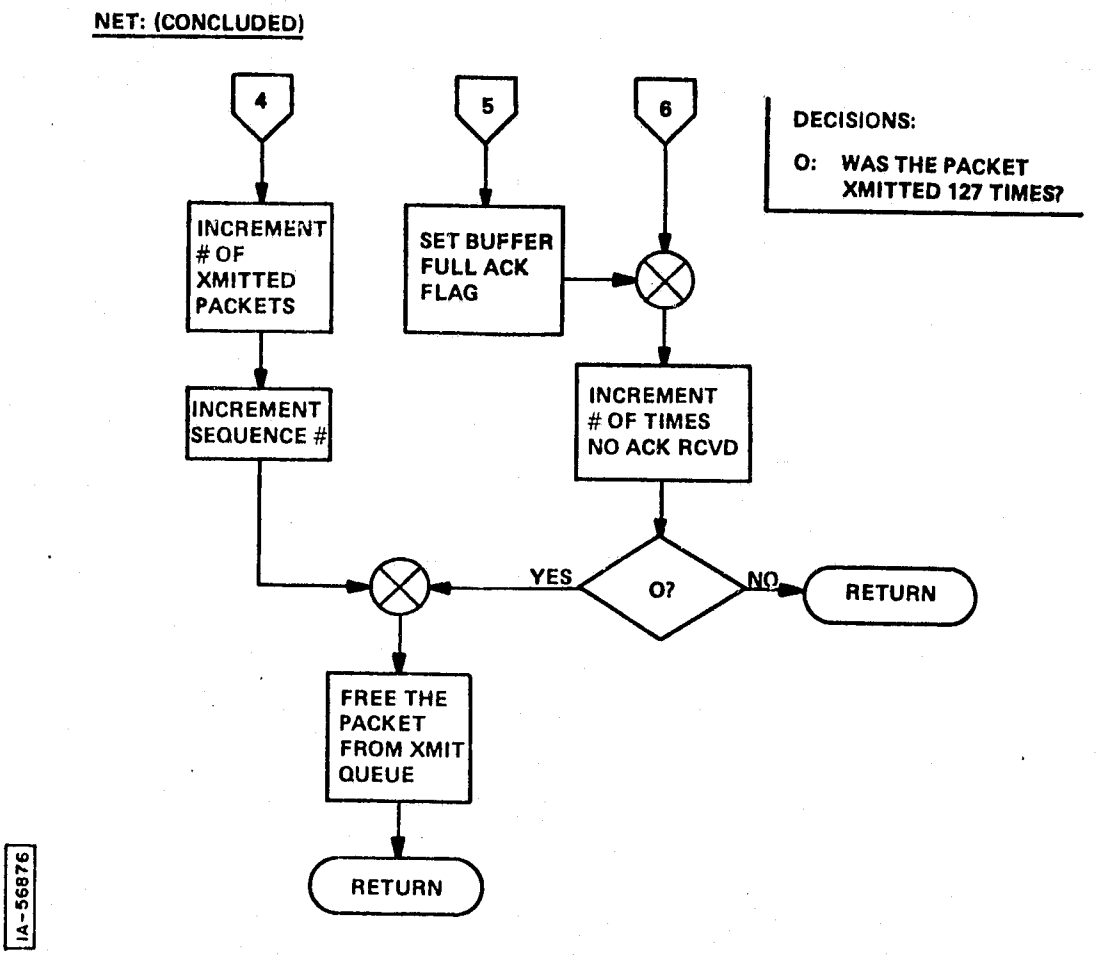


Figure 7.3-1: (Concluded)

7.3.1 Collision Avoidance

As mentioned in Section III, the network uses a system of collision detection and avoidance known as contention LWT. Should a collision occur during the transmission of a data packet, the XMIT flag is cleared and a pseudo-random number is loaded into RNDCNT. This variable is used to hold the transmission process in a delay loop to allow the network time to clear. Once the delay loop decrements RNDCNT to zero, the routine is allowed to continue normal processing. Since a random number based on the BIU's address is used to determine RNDCNT, the possibility of a deadlock situation, where two BIUs are continually colliding with each other's packets, is eliminated.

7.3.2 Flow Control

Since each BIU has a limited number of buffers available for storage of data packets, it is necessary to halt transmissions of additional data packets if all of the buffers are in use. This flow control management is accomplished by using the normal message acknowledgment function of the network protocol. When a receiving BIU exhausts all of the available buffers, it will ACK the next packet with $FF (See IRQ Routine Paragraph 7.11). The NET routine recognizes this ACK as a request for time to clear one or more buffers and enters a delay loop to allow time for the receiving BIU to catch up (i.e. clear some buffers). This is accomplished by loading the NOXMIT counter with $FF or a count of 255. Each pass through the LWAIT loop sets the RNDCNT counter to $32 which results in a 0.2 millisecond (msec) await. The total wait for an $FF ACK is thus approximately 51 msecs.

7.3.3 Listen-While-Talk

Once the required delays have been accomplished, the NET routine checks the status of the network ACIA (UART) to determine if another BIU is transmitting. If the $\overline{\text{DCD}}$ signal is low, the routine returns control to the MAIN LOOP for a retry at a later time. If the $\overline{\text{DCD}}$ signal is high, the network is free for the moment. At this point the routine initializes several variables to prepare for transmission of a data packet. Since a finite time delay exists between the end of a packet's transmission and the return ACK from the receiving BIU, it is necessary to allow sufficient time for this return ACK to occur. In the CAPS application this time delay is 100

microseconds (usecs). To avoid colliding with another BIU's ACK, the NET routine checks the network status again after a delay which allows ACKs to clear. If the network is found to be clear at this point, the transmitter carrier is turned on to keep other BIUs from transmitting and the data transmission is initiated. The XMIT flag is set at this point to detect the occurrence of a collision during the packet transmission loop.

7.3.4 Packet Transmission

The next section of the NET routine transmits the data packet on the network, one byte at a time, until the packet byte count has been satisfied. The loop used to transmit the data continually monitors the state of the XMIT flag to determine if a collision has occurred. If the XMIT flag is cleared before the completion of the data packet transmission signaling a collision, the routine halts transmission, increments its collision counter and returns to the MAIN LOOP and attempts retransmission of the packet at a later time. Each byte of the data packet is exclusively-ORed with all other bytes of the packet using the variable PARITY. After the packet byte count has been satisfied, the variable PARITY is transmitted on the network to serve as the longitudinal parity byte.

7.3.5 Acknowledgment Control

Each successfully transmitted packet should be acknowledged by the receiving BIU within a given time period. As mentioned above, for the CAPS application, the ACK must be received within 100 usecs of the parity byte. The NET routine enters a delay loop at the conclusion of the data transmission to await the packet's ACK.

7.3.5.1 No ACK. If the delay expires before an ACK packet is detected, the routine assumes the destination BIU did not receive the packet, increments the retry counter, NTPRX, and checks to see if the packet was sent 127 times. If it was, the lost packet counter, NMLOST, is incremented and the packet discarded to prevent the delay of other data packets in the output queue. If not, then control is passed back to the MAIN LOOP for another try later.

7.3.5.2 Bad ACK. The receipt of an ACK is detected by the receive data register full bit (bit 7) in the network UART status register being set. Since the UART is initialized to generate an interrupt when this condition occurs, it is necessary to disable CPU interrupts while waiting for the ACK. This is to prevent the processor from branching to the IRQ routine when an ACK is detected. When the status bit is set, the NET routine compares the data byte in the receiver to the address of the packet's destination BIU. If they do not agree, the byte is compared to $FF to determine if the transmitting BIU should delay for 50 msecs before trying to retransmit the packet. If it is not an $FF ACK, it is assumed to be a bad acknowledgment and the NTPRX counter is adjusted and the retry code executed as if no ACK was received.

7.3.5.3 FF ACK. When an $FF ACK is detected, the NOXMIT flag is set by storing the value $FF into it. This is the numerical equivalent of placing a count of 255 into the counter. At this point the code continues through the section dedicated to counting the number of times a given packet was retransmitted whereupon it returns to the MAIN LOOP until a later pass through NET results in the generation of the delay requested.

7.3.5.4 Good ACK. When the comparison of the received data byte indicates a good acknowledgment, the status message field NMXMT is incremented to indicate the successful transmission of a packet and the buffer freed from the output queue using DQ.

7.3.6 Freeing Packets

To release buffers to the free buffer pool, the NET routine calls the DQ subroutine when the packet has been successfully transmitted or when it was transmitted 127 times with no good acknowledgment, whichever occurs first. Upon completion of the dequeuing process in the DQ subroutine, the retry count is reset to zero and the routine passes control back to the MAIN LOOP.

7.4 INDEV Subroutine

INDEV is the main routine used to process data from the device for transmission on the network through the NET sub-

routine. Since the structure of this routine is highly dependent on the type of device the BIU is interfaceing, there is very little that is similar from one application to another. In the case of the CAPS application, the routines were developed from the ground up to interface with the Harris UBC and Sanders Graphic 7 systems. However, because the cable system was designed under the "cut and insert" philosophy, some sections of the INDEV routine used by the Harris BIU are similar to the INDEV routine used by the Graphic 7 BIU. Similar sections of the two routines are associated with the DMA/TTY determination plan, the character/-word transmission and the queuing scheme. Details on the routines as a whole may be found in Sections VIII and IX.

7.4.1 DMA Verses TTY Determination

During the FFLOOP initialization in the RESET routine, the DMA indicator flag, DMATTY, is set to the quiescent -1 state. Whenever the first input character from the device is detected, two timers are set (DMATM and TTYTM) and the DMATTY flag set to +1 to indicate a possible DMA operation. After the first character is processed, the code enters a wait loop pending the receipt of the next character from the interface. If no characters are received before the DMATM timer expires (approximately 650 usecs), DMATTY is decremented to indicate a possible TTY operation. In this case, a check is made to determine if a TTY control character has been received and should be sent immediately. If no control character was processed, the wait is continued until the next character is received or the one second TTYTM expires. In the case of the TTYTM expiring, the processed character(s) are transmitted and a new packet is started. The main purpose of this interface control is to decrease the amount of transmission overhead that would be required if only the DMA timer were used. That is, if only a DMA timer were used, each TTY character would result in a data packet being sent. This condition would exist because under normal conditions teletype characters would be available at the interface hardware at a much slower rate (greater than 1/4 sec. between characters) than a DMA operation. DMA operations are assumed to present characters to the interface within 650 usecs of each other. Each character presented resets the TTYTM flag and, if a DMA operation is assumed, resets the DMATM flag. If more than one word was processed via the DMA interface when the DMATM timer expires, the buffered data is transmitted. Whenever the buf-

fered data is transmitted in the INDONE section of the code, the DMA and TTY timers are reset and the DMATTY state flag returned to the quiescent -1 condition.

7.4.2 Character/Word Transmission

As characters or words are presented to the device interface, the appropriate handshake lines are set to insure no information is lost. While the names of the various handshake lines are different, the functions performed in the Harris BIU and the Graphic 7 BIU are the same. Upon entering the INDEV routine from the MAIN LOOP, a check is made of these handshake signals to determine if a valid data word is available on the data bus. If the handshake signal [on 6522(A)] indicating a data word is available has undergone an active transition, the appropriate handshake bit in P1IFR is set and the data word can be read (low-order byte on PORT1A and high-order byte on PORT1B). The loading of the CPU's "A" register from PORT1A triggers the return handshake signal to the device completing the handshake sequence. Due to the type of hardware used in the BIU design, the data presented to the BIU on the data bus is logically inverted. To obtain the proper format for transmission on the network, each byte is exclusively-ORed with $FF as it is processed.

7.4.3 Queuing a Packet for Transmission

The first check made upon entering the INDEV routine insures that a data packet is available for the receipt of data words from the device. The INSET flag is set whenever there is a buffer ready to receive data. If INSET is cleared, a new buffer is requested by using the PCONST subroutine. The buffered data is transmitted to the output queue whenever one of the following conditions exists:

1. The maximum packet length occurs.
2. Both the TTYTM and the DMATM expired.
3. A DMA operation of more than two words occurred and DMATM expired.
4. A TTY operation occurred and the last character was a control character.

Any of the above conditions causes the transfer of control to the INDONE section where a check is made to insure that NULL packets, that is, data packets containing no data (byte count < 8), are not transmitted on the network. At this point, the buffer is added to the output queue by invoking the ENQ subroutine. INSET is cleared to force another call to PCONST upon return to INDEV. From here control is passed to the top of the INDEV routine to allow continued processing of DMA data, if necessary.

7.5 OUTDEV Subroutine

As in the case of the INDEV routine, the OUTDEV routine is highly dependent on the device to which the BIU interfaces. In the CAPS application, however, the OUTDEV routines used in the Harris and Graphic 7 BIUs do share some similarities. These similar sections include the buffer initialization code, packet transmission scheme, some special message handling and the buffer freeing code. These similar sections are discussed in the paragraphs below.

7.5.1 Buffer Initialization

The first check made when entering the OUTDEV routine is to determine if there is a data packet to be processed. This is accomplished by examining the input queue pointer, QIN. If the pointer contains a value of -1, the end-of-queue marker, there are no packets awaiting transmission to the device. If, however, there is a valid pointer value in QIN, the buffer address indicated by this value is stored in OUTPTR for processing. At this point a determination is made concerning the purpose of the packet. If it is a data packet, it must be stripped of the header and transmitted to the device. This is accomplished in the section labeled PDATA. If, however, it is not a data packet, the packet must be processed by special handlers appearing later in the routine.

7.5.2 Packet Transmission

In general, the same type of data communication and handshaking procedure is used in OUTDEV as was used in INDEV. That is, a data word is loaded on the data bus and a handshake signal [on 6522(B)] used to indicate that fact. The handshake and data lines are held in this state until the device responds with the appropriate counter-handshake signal.

In OUTDEV, the BIU originates the handshake by storing a data byte in PORT 2A. At that point a tight loop is entered to await the return handshake signal on the PORT 2 CA1 input which registers in P2IFR. The initialization of the PORT 2 VIA established a level handshake on CA2 which is cleared by the active transition of the CA1 signal line. This clearing action by the return handshake (DATU or OWR) from the connected device insures that the data ready handshake (DAVFU or OCTL) falls. Upon completion of the handshake sequence, control is returned to PDATA to continue transmitting data words until the packet's byte count is satisfied.

7.5.3 Special Message Processing

For the most part, the Sign-On and Sign-Off message handling sections are similar from one application to another. In the CAPS BIUs, however, some minor differences do exist. They are similar in regard to the setting and clearing of the appropriate flags used to control the state of the BIU; however, there are differences in the handling of Sign-On acknowledgments.

7.5.3.1 Sign-Off Message. If a Sign-Off message is being processed, the first check is to see if this BIU is connected. If it is not, the packet is ignored. If it is connected, the XADDR variable is compared to the from address in the packet. This is to insure that the BIU ingnores Sign-Off packets from BIUs other than the one to which it is connected (See Paragraphs 8.1.3.1 and 9.1.3.1). If the address is the same as the value in XADDR, the XADDR variable is cleared along with INBC. This is to insure that a partially generated outgoing packet is not inadvertently sent to the wrong address. After these two variables are cleared, the CONECT flag is reset to -1.

7.5.3.2 Sign-On Message. The processing of a Sign-On message also requires a check of the connection state of the BIU. If the BIU is already connected to another BIU, it cannot allow another BIU to Sign-On unless it is a special multifunction unit; therefore, the packet is ignored. In the CAPS application, neither the Harris nor the Graphic 7 BIUs allows more than one simultaneous logical link. If the BIU is not currently connected, the packet is examined for the return address. This value is then stored in XADDR and a

Sign-On Acknowledgment packet generated in PCONST. This packet is then added to the output queue by a call to ENQ and the connect flag CONECT double-incremented to +1.

7.5.4 Packet Freeing

As in the case of the NET routine, when the OUTDEV routine is finished processing a packet, a call to DQ is used to free the packet for later use.

7.6 TIMOUT Subroutine

The TIMOUT routine is used to increment a three-byte time-of-day variable to record the passage of 1/4 second intervals. As mentioned above in the RESET routine, the VIA clock in PORT 1 is used to mark 1/100 second intervals by setting a bit in the PIIFR. Each time the MAIN LOOP transfers control to the TIMOUT routine, that bit is examine and reset, if necessary. When it is reset, a variable labeled TICK, initialized to 25, is decremented until it reaches zero, indicating the expiration of 1/4 seond. When TICK reaches zero, it is reset to 25 and the variable TOD incremented via a three-byte addition scheme. Figure 7.6-1 is a flow chart of the operation of this routine.

7.7 CKTOUT Subroutine

Since the CKTOUT routine is used to indicate the state of several timers that are used to control certain functions of the BIU, there can be differences in this code to reflect the tests that are required by a given application. In the CAPS application, there are some differences between the Harris and the Graphic 7 BIUs in this area, but the basic scheme used to check for timeouts is the same. This routine functions by loading a pointer to the particular timer that is to be examined and then jumping to the CTIME subroutine. Upon return from that subroutine, the carry bit in the CPU's status register will be set to indicate the timer has expired or it will be cleared to indicate there is still time remaining.

C-2

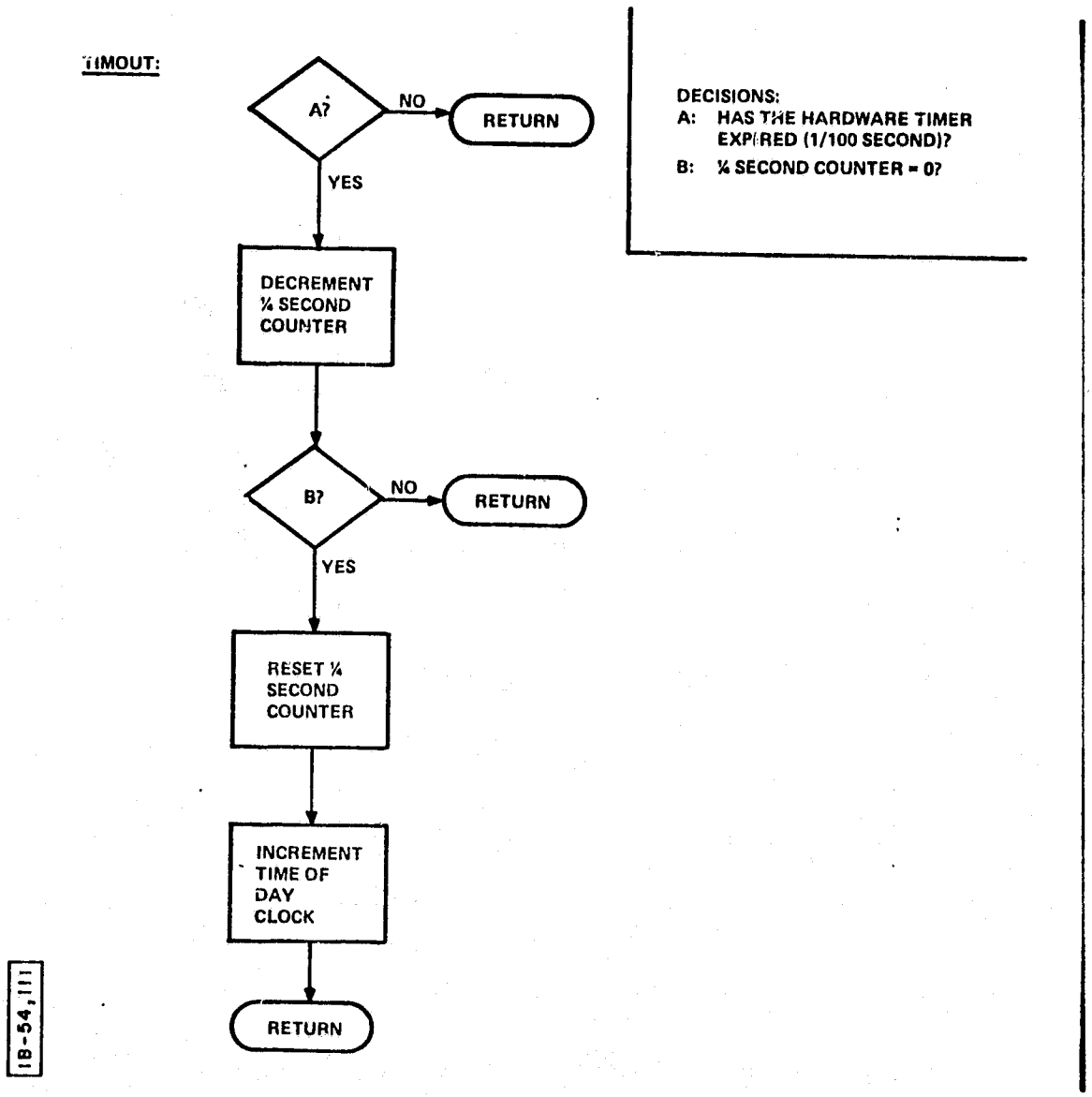


Figure 7.6-1: TIMOUT Subroutine Flow Chart

7.8 CTIME Subroutine

This routine is used to compare the current TOD variable to the variable indicated by the "X" register. It performs a three-byte check to determine if the timer has expired. If the TOD variable is greater than or equal to the variable being checked, the carry bit will be set when control is passed back to the calling routine.

7.9 SNDSTA Routine

Every BIU, that interfaces a subscriber device to the network, is designed to report the status of certain key parameters involving the network operation on a periodic basis. In most applications, this report is generated approximately once every minute. In the CAPS application, this status reporting is in the standard BIU format described in Section III of this report. The SNDSTA routine is used to generate this report whenever the CHKOUT routine indicates that the status message timer, TSTAT, has expired. The routine uses several loops to load a status packet with the data contained in the status variables. This status packet is generated by a call to PCONST with the special status message type code loaded into the "A" register.

7.9.1 Main Status Variables

After the status message packet length of 42 bytes is stored in the byte-count location in the packet, the routine enters a loop to transfer the first 16 bytes of status information into packet byte locations 8 through 23. This is accomplished by the section labeled SLOOP1. The bytes are obtained from the 16 consecutive bytes of page zero memory starting with the variable NMXMT and ending with the variable NMWAIT.

7.9.2 Packets Waiting

The next loops encountered in this routine are used to count the number of packets that are waiting to be transmitted to the device or transmitted on the network. In SLOOP2, the QOUT pointer is examined to start a count of the number of packets waiting to be transmitted onto the network. This loop increments a counter, the "A" register, until it encounters the end-of-queue marker, a -1, in the NEXT list.

This counter is then an accurate indication of the number of packets remaining to be transmitted. In a like manner, SLOOP3 is used to examine the input queue to determine the number of packets waiting to be transmitted to the connected device.

7.9.3 ACIA (UART) Status

The next few lines of the code are used to record the status of the device and the network UARTs. Here, also, the status of the interrupt flag register for the PORT 1 VIA is recorded along with the address of the last BIU to which a packet was sent (XADDR). Note that to enable a standard status packet format for all BIUs, the status of PORT 2 and PORT 3 VIAs are not examined and the device UART status is transmitted even though it is not used in the CAPS application.

7.9.4 BIU Function Code

The final 12 bytes of information loaded into the packet are transferred from an area of the EPROM which provides a description of the function performed by this BIU. In the CAPS application, the two codes used are "HARRIS BIU " and "GRAPH 7 BIU ". These last bytes are transmitted to the packet in the section labeled SLOOP4. At the conclusion of this loop, the packet is ready to be placed on the output queue. A call to ENQ is used to accomplish this.

7.9.5 Clearing Counters

The final loop, in this section of the code, is used to clear the 16 counters used to record the status message data described above. In SLOOP5, zero is stored in the 16 page zero locations starting with NMXMT. Following this loop, control is return to CKTOUT to finish the timer checks.

7.10 INTBUF Subroutine

The INTBUF routine is used to insure that a packet is ready to receive data from either the network or the attached device. A flag, INTSET, is used to indicate when a buffer is ready. Upon entering this routine, the INTSET flag is examined. If it is cleared, then a new buffer must be prepared

to receive data. This buffer initialization is accomplished by a call to the ALLOC subroutine. If control is returned to the INTBUF routine with the negative bit of the CPU status register set, then the buffer allocation was unsuccessful and control is returned to the original calling routine. If the allocation was successful, then the input buffer pointer, INTPTR, is set with the buffer address and the INTSET flag is incremented. The flow chart in Figure 7.10-1 provides a logical picture of the functioning of this routine.

7.11 IRQ Subroutine

As mentioned in Table VII, this routine is initiated by a hardware vector. This vector is located at memory location $FFFE. Whenever the CPU's IRQ signal goes low, the program counter is loaded with the address of the IRQ subroutine from that vector location. Since this routine is another of the network interface routines, most of the code will be common to all BIUs in the application. There is one small area of difference between the Harris and the Graphic 7 BIUs. The Harris BIU must recognize a DISC signal from the UBC as an interrupt, therefore several lines of code were added to the standard IRQ routine to handle this function. This special code is discussed in Section VIII. The flow chart shown in Figure 7.11-1 is useful in understanding the functions of this routine.

7.11.1 Interrupt Initialization

To prepare to process an interrupt, the IRQ routine must first insure that no information concerning the state of the CPU can be lost during the routine. For this reason, the state of each register is pushed onto the hardware stack. Note that, as part of the interrupt vectoring process, the CPU status register and the return address for the program counter were previously pushed onto the stack.

7.11.2 Receiver Control

The network UART was initialized to generate an interrupt whenever any one of the following three conditions exists:

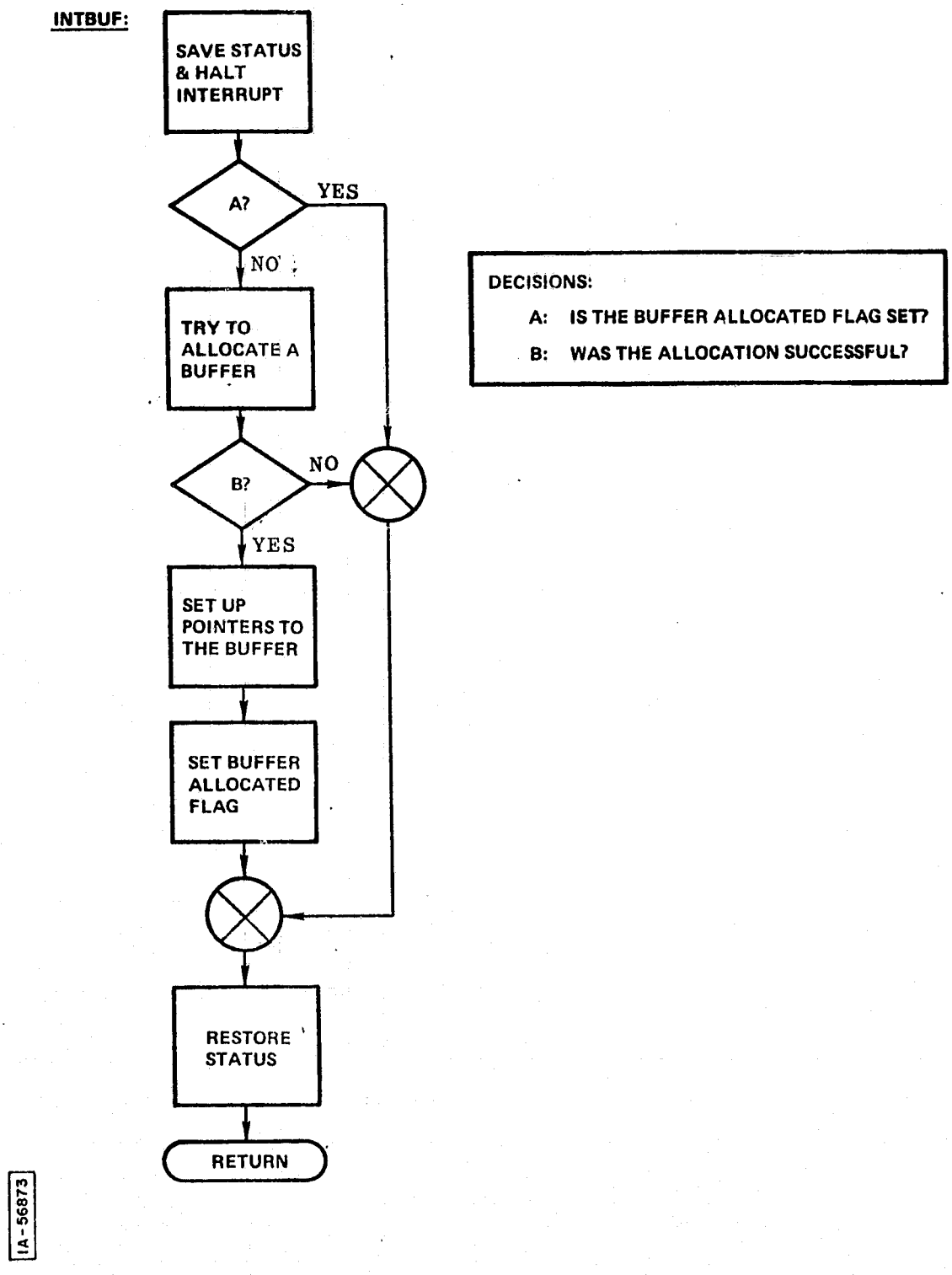


Figure 7.10-1: INTBUF Subroutine Flow Chart

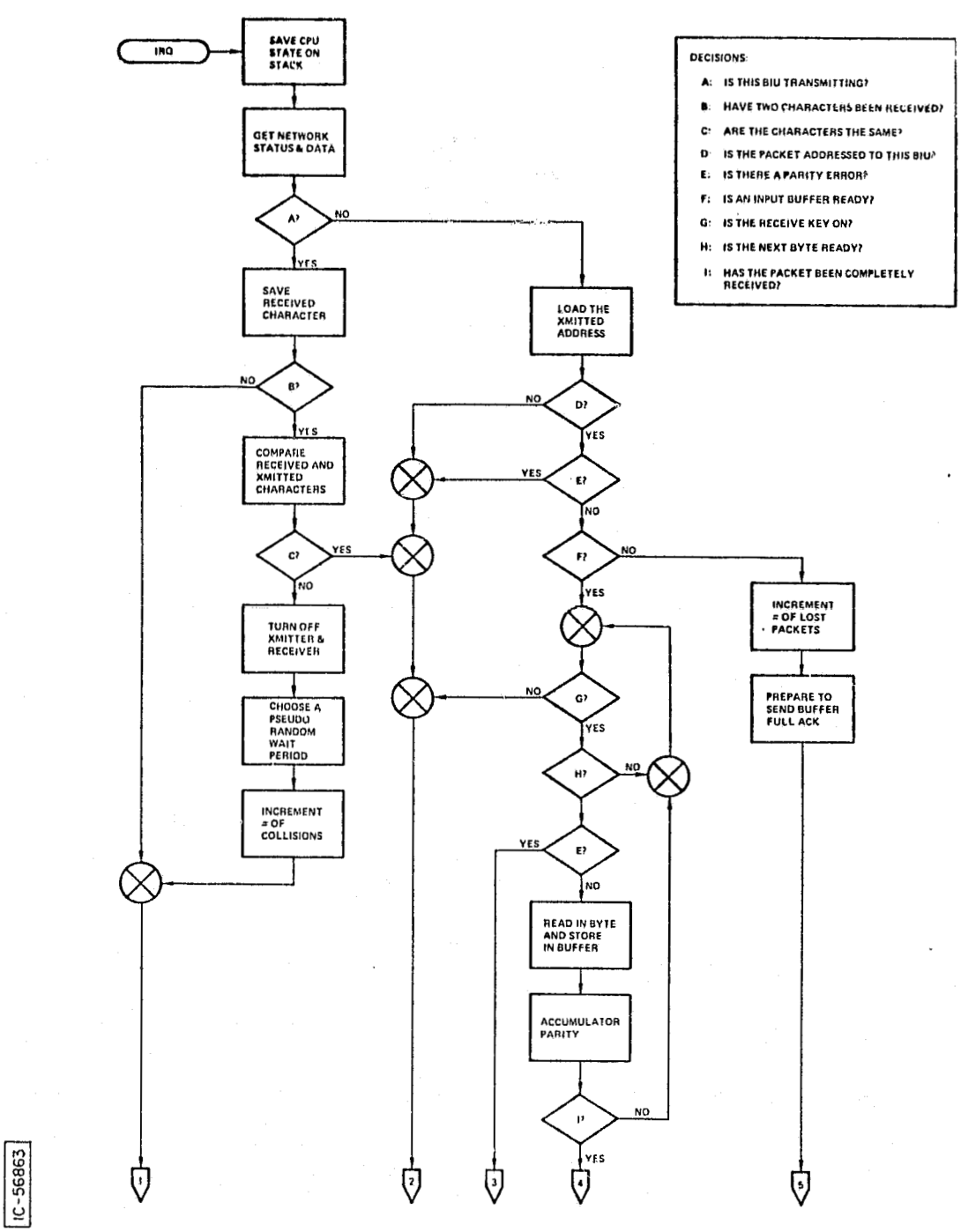


Figure 7.11-1: Standard IRQ Routine Flow Chart

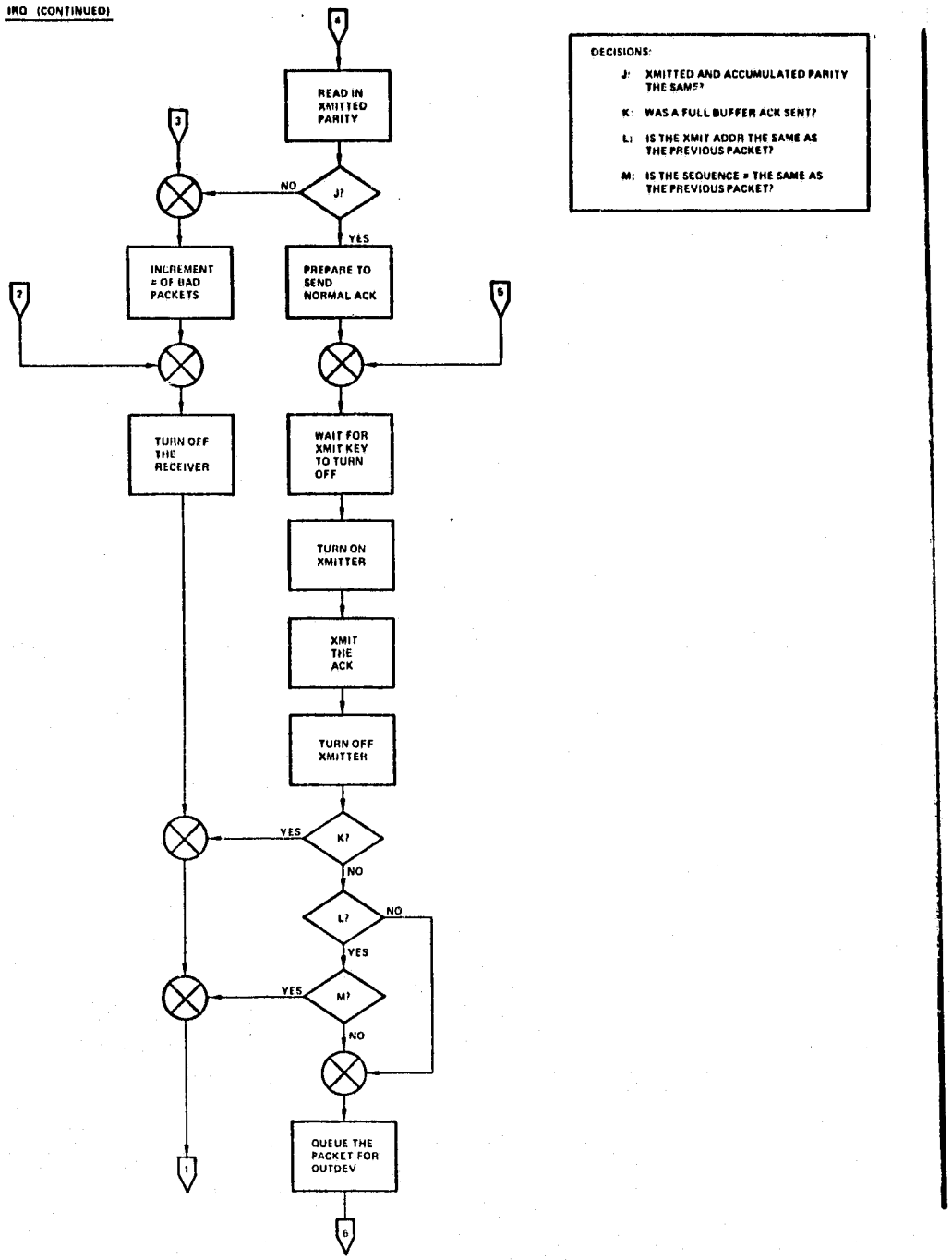
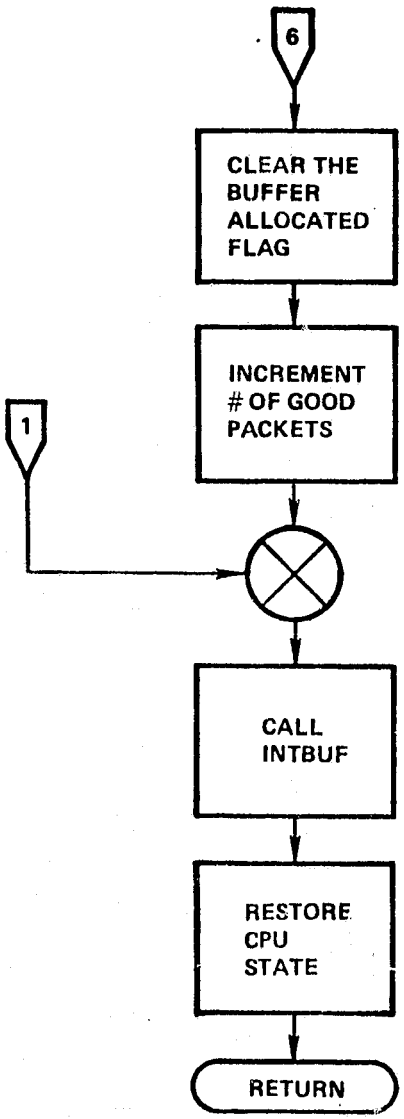


Figure 7.11-1: (Continued)

IRQ: (CONCLUDED)



IA-56874

Figure 7.11-1: (Concluded)

1. Receive Data Register Full

2. Receive Overrun Condition (No Stop Bit)

3. $\overline{\text{DCD}}$ Transitions Low-to-High

To clear the overrun condition, the network UART status register is loaded into the "Y" register. The receiver data byte is then placed in the "A" register and the determination of the destination of the data packet is initiated.

7.11.3 Data Destination

Two possible conditions can exist when the receiver detects a date byte on the network. First, the packet could be one that this BIU is transmitting using the LWT protocol, or second, the packet could be coming from another BIU. The XMIT flag is used to distinguish the two conditions.

7.11.3.1 Own Transmission. If the XMIT flag is non-zero, the data byte in the receiver is the result of one of the first two bytes transmitted by the NET routine. These are the bytes that are used to determine if a collision with another BIU's transmission has occurred. The first two bytes are stored temporarily in the two byte variable TOKEN to await the collision comparison. If the two bytes agree with the first two transmitted by NET, then the receiver is turned off to allow uninterrupted transmission of the remainder of the outgoing packet. If a collision was detected, both the receiver and the transmitter are turned off, resulting in the clearing of the XMIT flag (See NMI Routine Paragraph 7.12) and a pseudo-random number is generated and stored in RNDCNT to delay the next attempt at packet transmission. After the setting of RNDCNT, the number of collisions, NMCOL, is incremented and control passed to the return section of the IRQ routine. Following the return from the interrupt, control is passed back to the NET routine where the absence of the XMIT flag is detected during the transmission loop and transmission halted.

7.11.3.2 External Transmission. If the XMIT flag is zero, the byte in the receiver buffer must be coming from another BIU. If this is the first byte of a packet, it contains the address of the destination BIU. In all current bus

applications, only the first byte of the packet is examined for this addressing information, though, as was noted in the discussion on the packet structure in Section III, two bytes are provided for addressing. In the CAPS application, the first byte is compared to an address selected via hardware on PORT 3B. This function is described more fully in Section V. If the packet is not addressed to this BIU, the receiver is turned off until the falling of the transmission carrier generates a Non-Maskable Interrupt (NMI).

7.11.4 Data Reception

If the packet is addressed to this BIU, a check is made to see if there is a buffer ready to receive the data. INTBUF is examined, and if found to be cleared (indicating that a buffer is not ready), the number of lost packets, NMLOST, is incremented, the receiver turned off and an $FF ACK transmitted to the originator BIU. If a buffer is available, the address is stored in the first buffer memory location and the input loop entered. At each reception of a data byte, the byte parity is checked. If there is a parity error detected, control is passed to the parity error handler section of the routine. In addition to the byte parity check, each data byte is exclusively-ORed with the variable PARITY to repeat the longitudinal parity check discussed in NET. When the eighth data byte is detected, it is stored in the special byte count variable, INTBC, to allow the determination of the end of the packet. Once this byte count is satisfied, the next byte received is the longitudinal parity byte. This byte is then compared to PARITY and control passed to the acknowledgment section if it is the same value.

7.11.5 Parity Error

If the longitudinal parity byte is in error or if any byte parity bit is in error, control is passed to PARERR. Here the number of packets received with bad parity, NMBCRC, is incremented, the receiver turned off and control passed to the interrupt return sequence without sending an ACK. Since no acknowledgment is sent, the originating BIU is forced to retransmit the packet at a later time.

7.11.6 Acknowledgment

This section of the routine can be entered from two areas of the code. If a packet was received without error, the code transfers control to SNDACK with the BIU's address in the ACKBYT variable location. If there were no buffers available for reception of the packet, the value in ACKBYT is set to $FF. This section of the code waits until the network becomes free and then transmits this one byte as an acknowledgment to the originator of the packet.

7.11.7 Retransmissions

To allow for the possiblity of a data packet being processed more than once, as in the case of the originator missing an ACK, each packet header contains a sequence number. The originator's address and the sequence number of the packet are examined to determine if this packet had been received previously. The two bytes in the packet are compared to the values stored in LASTTX and LASTSQ and if found to be the same, the packet is ignored. If it was not received earlier, the new sequence number and originator address are saved in LASTSQ and LASTTX respectively.

7.11.8 Packet Queuing

All new packets are added to the input queue in a manner similar to that used for queuing packets for transmission in INDEV. The input queue is specified by the value QIN-NEXT and the ENQ routine is called. The number of packets received with good parity, NMGCRC, is incremented and a new input buffer requested by a call to INTBUF. At this point control is passed to the interrupt return code.

7.11.9 Interrupt Return

The process of saving the state of the CPU registers is reversed in the interrupt return code. Here the three register values are pulled from the hardware stack and an RTI command executed. The RTI command insures that the CPU status register is restored from the stack as well.

7.12 NMI Routine

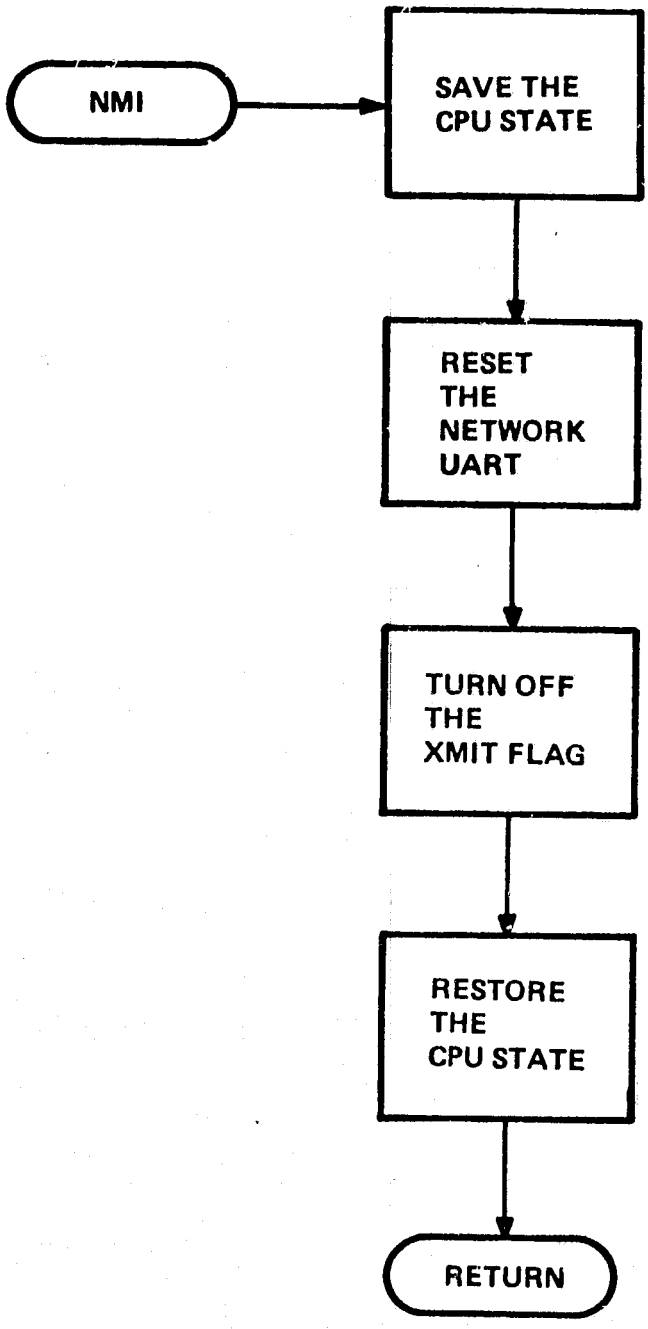
A non-maskable interrupt vector is located in the EPROM memory at location $FFFA. This interrupt is used as an indication that the data carrier on the cable has fallen. This active transition of the carrier signifies the end of a BIU's transmission. The NMI code is used to reinitialize the network UART, that is, turn on the receiver to counteract the effects of the IRQ routine when it determines that a given packet is not of interest. It is also used to clear the XMIT flag thus signifying that a collision may have occurred. The standard interrupt handling procedure for saving the status register and the return address is used and the "A" register pushed and pulled as required to save its state. Figure 7.12-1 is the flow diagram for this routine.

7.13 PCONST Subroutine

This subroutine is used to prepare a packet for transmission onto the network. The first step in this process is to allocate a buffer for this packet. Since the allocation process involves the adjustment of buffer pointers used in the IRQ routine, the ALLOC subroutine must not be interrupted, thus interrupts are disabled during the call to ALLOC. Here again the state of the negative flag in the CPU status register is used to indicate the success of a buffer allocation. In this case, however, if a buffer is not available, the routine simply returns control to the calling routine until a later time. If a buffer was successfully allocated, BIUPTR is set to the buffer address and the packet header initialized. CLOOP is used to clear the header values not set in the PCONST routine. The destination address is loaded into the packet from the value in XADDR and the BIU's return address is derived from the PORT 3B value. The message type was passed to the routine in the "A" register and is stored in the sixth byte of the header. A default packet length of 8 bytes is assumed at this point and control returned to the calling routine.

7.14 STIMER Subroutine

The STIMER routine is used to set the value of a three-byte system clock variable based on the TOD timer. The pointer to the clock to be set is loaded into the "X" regis-



IB-54,118

Figure 7.12-1: NMI Routine Flow Chart

ter and the value to be added to TOD is loaded into the "A" register. The routine then uses a three-byte addition scheme to store the TOD plus increment into the indicated timer.

7.15 SFINC Subroutine

This routine is used to increment two-byte status message fields in a non-return to zero scheme. It is called with the counter to be incremented indicated by the value of the "X" register. This counter is then incremented and, if the total value of both bytes of the counter exceeds 65535, the counter is set to $FFFF.

7.16 ALLOC Subroutine

As was mentioned above, the ALLOC routine is used to reserve a buffer for use in storing data coming into the BIU from the network or the attached device. ALLOC examines the value in STKPTR, the variable indicating the number of buffers not previously allocated, and if decrementing STKPTR does not result in a negative number, a buffer is allocated. A negative value in STKPTR at this point results in a direct return to the calling routine with the sign bit of the CPU's status register set. If there is a buffer available, the "X" register is set with the pointer to the free buffer and control is passed to the calling routine.

7.17 ENQ Subroutine

As in the case of the PCONST routine, the ENQ routine must make adjustments to buffer pointers. Therefore, the status of the CPU's interrupt flag is saved by pushing the status register on the stack and disabling interrupts. ENQ is called with the queue of interest, input or output, indicated by the value in the "A" register. The value indicated by this pointer is itself a pointer to the next value in the queue. By searching throught the NEXT list (a connected list of buffer pointers), the end-of-queue indicator, $FF, is found. When the $FF is located, the pointers are spliced to include the new buffer according to the scheme indicated in Figure 7.17-1. The interrupt status of the CPU is restored by the PLP command and control returned by the RTS command.

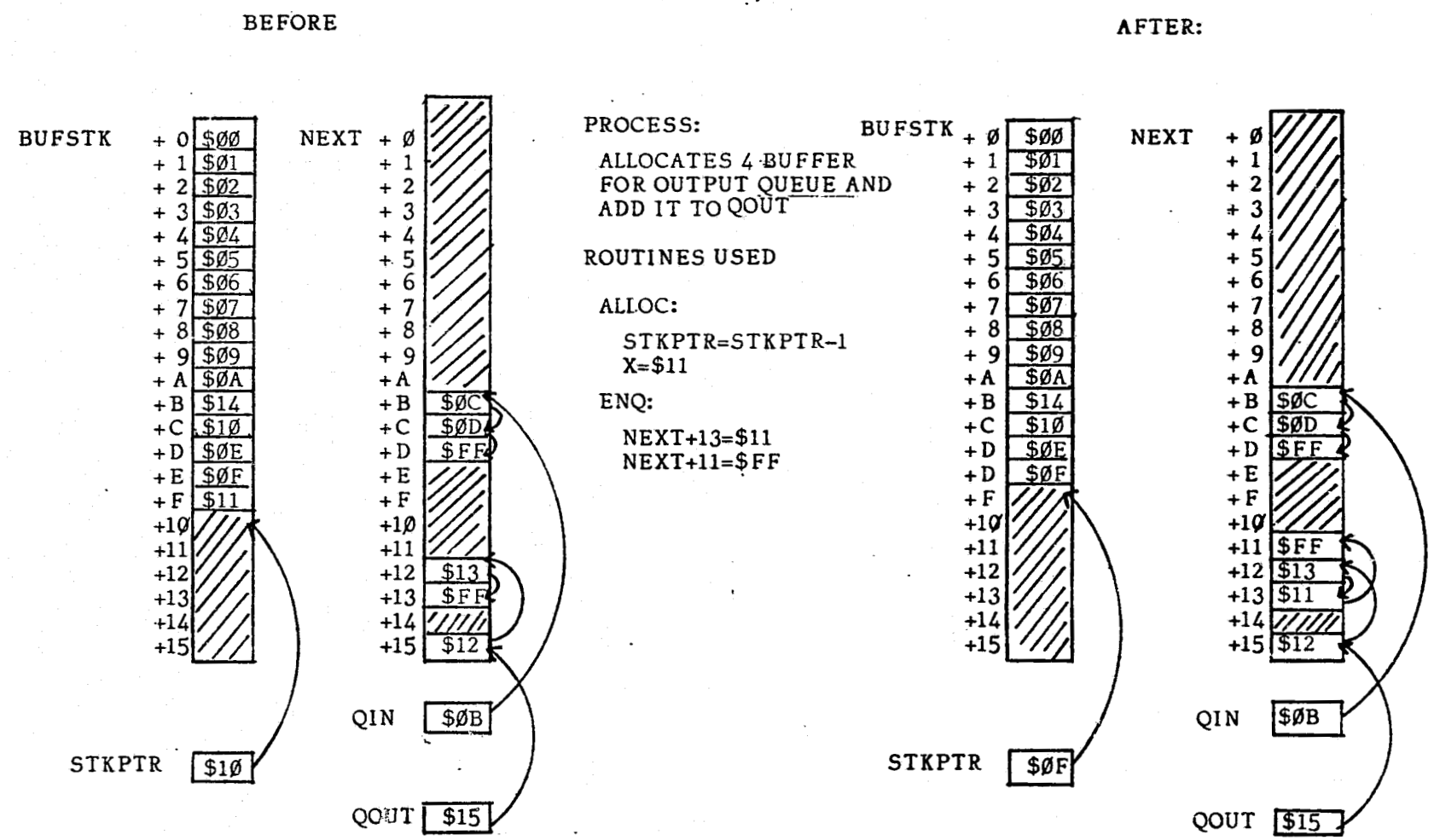


Figure 7.17-1: Packet Enqueuing Procedure

7.18 DQ Subroutine

As in the case of the ENQ routine, the DQ routine must not be interrupted by IRQ. Therefore, the interrupt mask is set after the interrupt state of the CPU is saved by the PHP command. The top buffer in the queue indicated by the "X" register contents, is removed by the adjustment of pointers indicated in Figure 7.18-1. The number of the free buffer is added to the list of free buffers in BUFSTK and the value of STKPTR is incremented to indicate one more buffer is now available. The interrupt status of the CPU is restored by the PLP command and control returned to the calling routine by the RTS command.

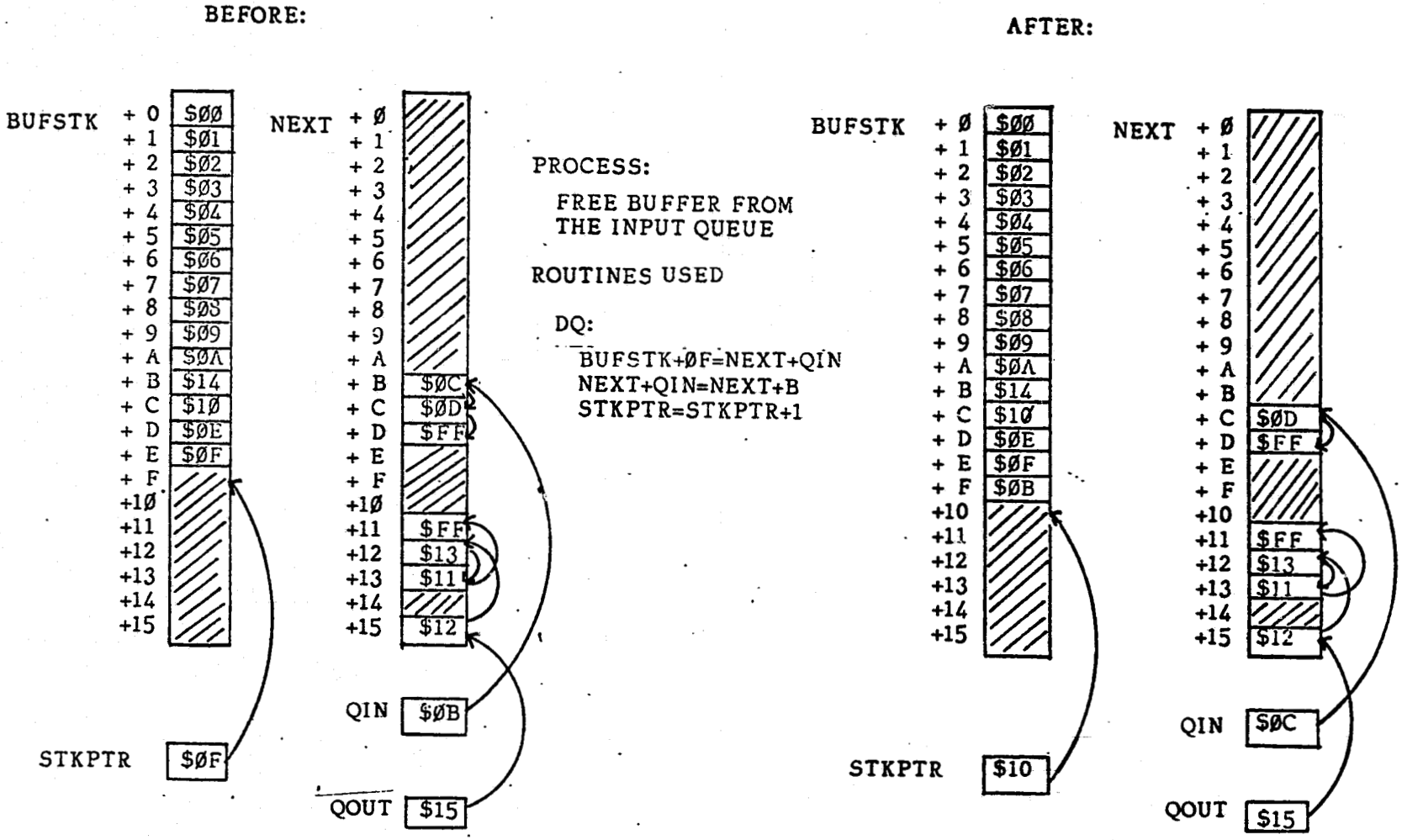


Figure 7.18-1: Packet Dequeuing Procedure

SECTION VIII

SPECIAL HARRIS UBC BIU SOFTWARE

8.0 INTRODUCTION

As mentioned in Section VII, some sections of the BIU software were designed for a particular device interface. The Harris UBC has several handshake signal lines which are not necessary on the Sanders parallel interface card. For this reason and because the BIU must resemble a Sanders Graphic 7 parallel interface to the UBC, there are differences in some sections of the BIU code. These differenes in the Harris UBC software are detailed in the following paragraphs.

8.1 RESET Routine

Although much of the RESET routine is common to both the Harris and the Graphic 7 BIU software, portions of the routine are dedicated to special initialization functions. These functions include handshake definition, setting the $\overline{R}/W$ state and preparing the BIU to respond to inputs from the VOLLEY diagnostic program (See Section X). The following paragraphs detail these special functions and Figure 8.1-1, a flow chart of the special Harris BIU RESET routine, is provided to allow the reader a clearer understanding of the discussions.

8.1.1 Handshaking Initialization

The 6522 VIA chips must be initialized to recognize proper logic levels or pulses on the CA1 and CB1 inputs and to establish the proper responses on the CA2 and CB2 outputs. In the Harris BIU, the handshaking controls are set according to Table IX. This table indicates the signal change that is recognized as an active transition by the 6522 and the data word stored in the chip's PCR to initialize it.

In addition to establishing the handshake control, this code also determines which signals are used to interrupt the CPU. In the case of the Harris UBC BIU code, the UBC's transmission of a DISC signal should cause the BIU to respond with a CNCT signal. For this reason, the PORT 3 IER is

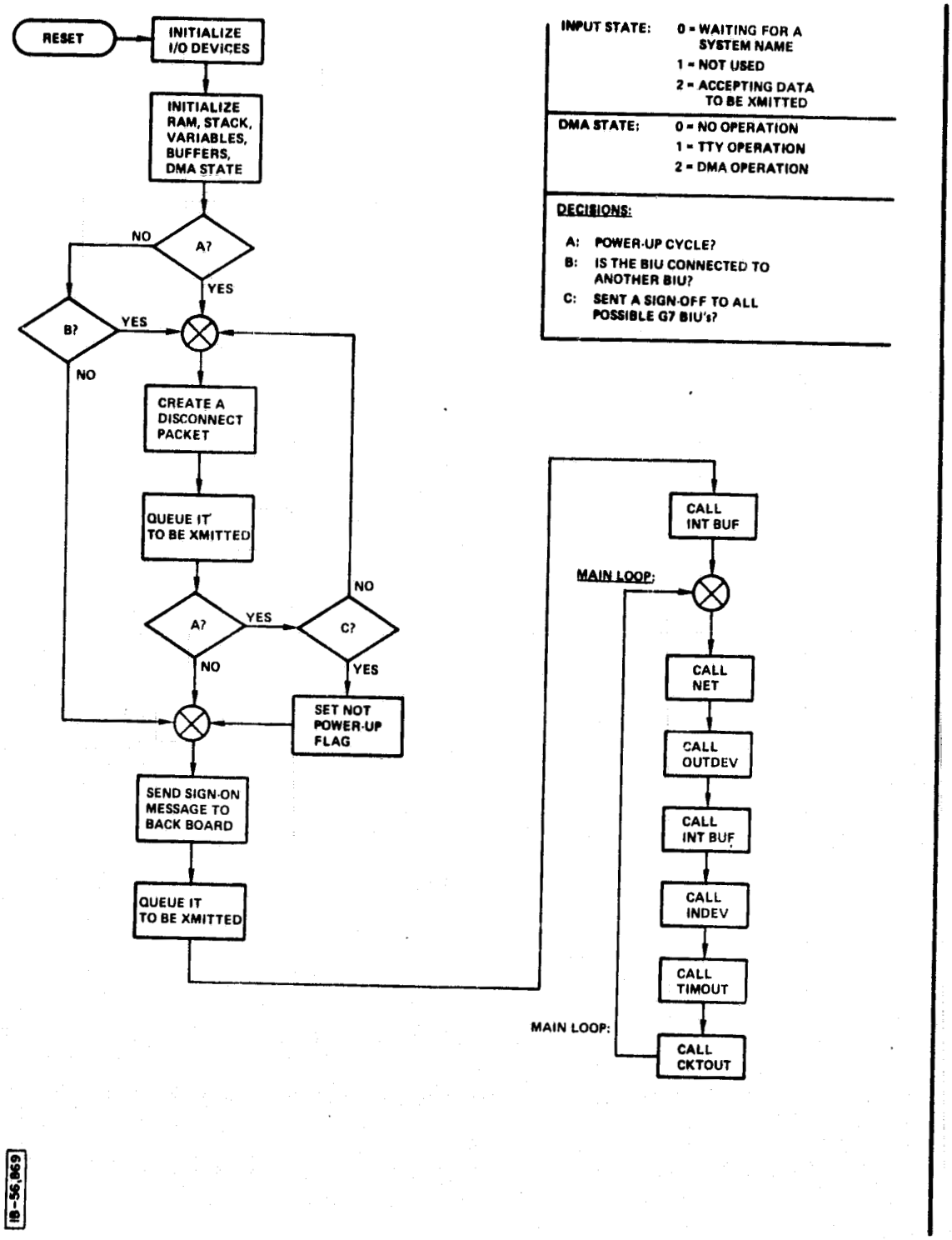


Figure 8.1-1: Harris BIU RESET Routine Flow Chart

loaded with the code 10000010 to indicate that an active transition on PORT 3 CA1 should generate an interrupt on the BIU's CPU $\overline{IRQ}$ line. Special code discussed in Paragraph 8.5 is executed when this occurs.

TABLE IX

HARRIS UBC BIU HANDSHAKING CONTROL

PORT #	DEFINITION	INIT CODE	ACTIVE TRANSITION CA1	ACTIVE TRANSITION CB1	RESPONSE CA2	RESPONSE CB2
1	INPUT	10011011	L-TO-H	L-TO-H	PUL	N/A
2	OUTPUT	10001000	L-TO-H	N/A	LEV	N/A
3	CONTROL	11111000	L-TO-H	N/A	MAN	N/A

L-TO-H = Low to High (0V to +5V)
PUL = 550 ns +5V Pulse
LEV = +5V Level held until active transition on CA1
MAN = Manual Handshake under software control
N/A = Not Used

8.1.2 $\overline{R}$/W Status

To insure that the status lines used by the Harris UBC to indicate to the communications handler which type of I/O operation generated the last IIFU signal, are in a known state, the special write data word is stored into PORT3A. With bit 4 of this word set to zero and the data inverted by the driver hardware, the last operation performed by the UBC is initiated as a write operation. This line is set manually throughout the I/O routines of the Harris code in the INDEV and OUTDEV subroutines.

8.1.3 Power-Up Cycle

Due to constraints on user transparency, the power-up or power-fail condition demands that the Harris BIU attempt to clear any possible Graphic 7 BIU connections and force the previously connected Graphic 7 BIU to reinitiate a sign-on to any available Harris BIU without user interaction. This necessitates a complex power-up sequence in the Harris BIU. With multiple terminals connected via the bus, the virtual

circuits created by the BIU interconnections are destroyed by a power failure or removal of power (turning the BIU off) at either end. To allow for a smooth reconnection of these circuits when power is restored, each BIU executing the RESET code issues a sign-off message to the other BIU comprising the old circuit. This is to prevent the possible loss of resources if one BIU assumed it was still in the connected state (See Paragraph 7.5.3.2). Since the previous state of the BIU is lost when power is removed (dynamic RAM), the Harris BIU sends a sign-off message to all possible Graphic 7 BIUs when executing the RESET code during a power-up sequence. This is not the case, however, when the RESET button is pushed. In this case, only one sign-off message is necessary and that is to the BIU whose address is still present in XADDR. In the case of a power-up cycle, the Harris BIU attempts to connect to the backboard BIU (See Paragraph 8.1.4) unless a Graphic 7 BIU sign-on request overrides this link. The criterion used to distinguish between the pressing of the RESET button and a power-up cycle is based on the loss of data due to the dynamic RAM. If the variable HARGO contains the bit pattern $57 then the code assumes this is not a power-up cycle and continues executing the code as if the RESET button was pressed.

The constants TRMBAS and TRMCNT are used to establish the limits for the sign-off loop. TRMBAS is used to indicate the base network address of all Graphic 7 BIUs. It should be the lowest address for any set of Graphic 7 BIUs and all other Graphic 7 BIUs should have an address which falls within the range between TRMBAS and TRMBAS+TRMCNT-1. TRMCNT is used to indicate the total number of possible Graphic 7 terminals in the system. In the current application, TRMBAS is set to $51 and TRMCNT is set to $08. The sign-off loop, SNDSOF, repeatedly calls PCONST with a new value of XADDR within the range described above and then calls ENQ with the newly defined sign-off packet. The final result of this loop is a queue of sign-off messages, one to each of the possible Graphic 7 BIU addresses in the system.

8.1.4 VOLLEY Initialization

A special diagnostic program was developed for the Harris Slash 8 to allow for the verification of the cable hardware and to troubleshoot cable problems (See Part IV). Special BIU software was developed for the Harris BIU to allow this program to communicate directly to the Backboard BIU whenever the Harris BIU is not already connected to a

Sanders terminal. After the connect state of the BIU is established during the RESET cycle, the BIU generates a sign-on request to the Backboard BIU. Control is then passed to the main loop to allow normal processing of the sign-on request. Code in the NET, INDEV and OUTDEV subroutines handles the completion of the sign-on process and the establishment of communication with another BIU when that BIU attempts a sign-on to the Harris BIU.

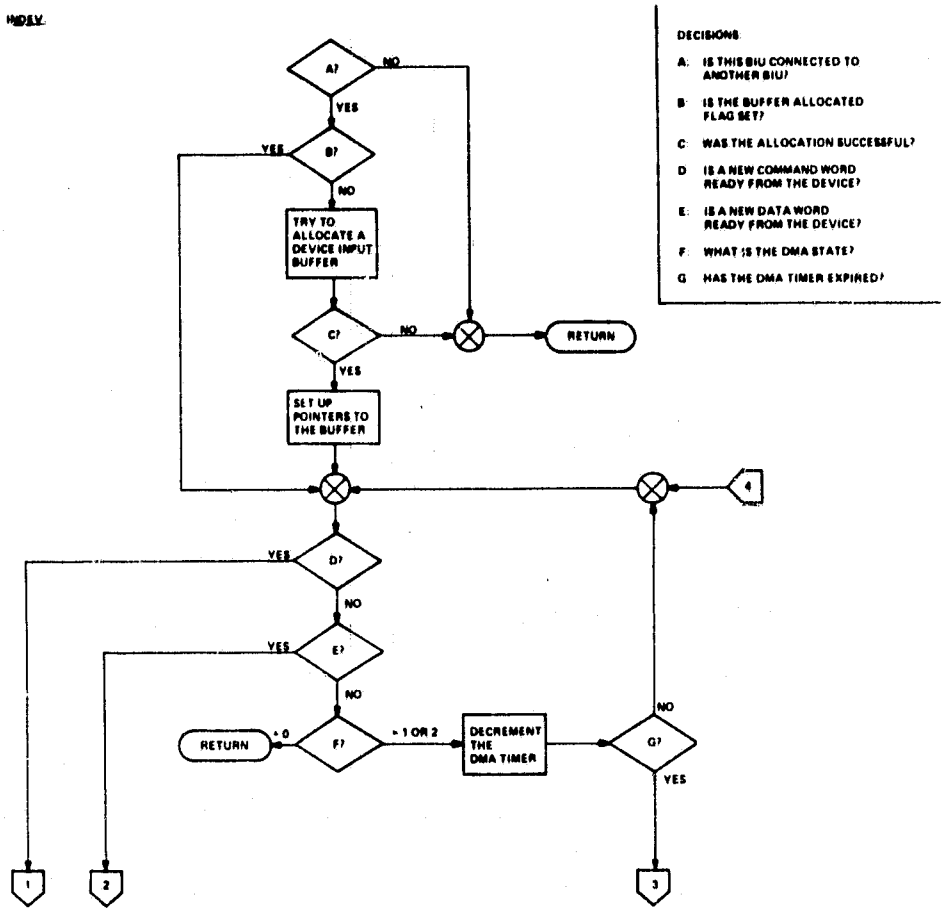
8.2 INDEV Subroutine

The INDEV subroutine must perform all of the necessary Harris output processing software functions of the Graphic 7 which the BIU mimics. In the case of the Harris UBC BIU, this requires INDEV to operate like the Graphic 7 parallel interface handler. Unlike the Graphic 7 BIU, however, the Harris BIU does not require the capability to enter into a dialogue with the attached device to determine to which BIU it will connect. For this reason, the INDEV routine used by the Harris BIU contains only code associated with the device interface. This special code involves several functions including handshaking, command data processing and interrupt generation. The following paragraphs detail these functions. A flow chart of the Harris INDEV subroutine is presented in Figure 8.2-1 for reference.

8.2.1 Harris Handshaking

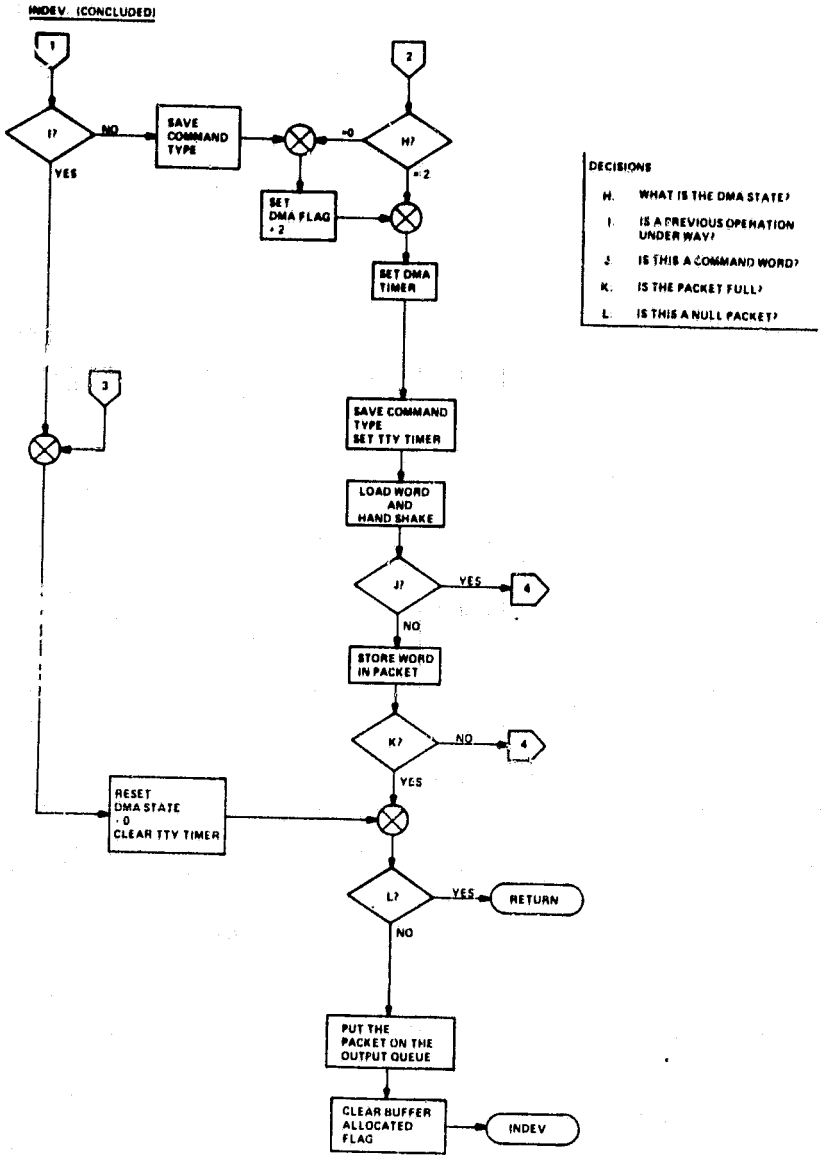
The Harris BIU handshaking signal lines must respond to the UBC in a manner similar to the Graphic 7 parallel interface. Since the PORT 1 VIA [6522(A)] is used for input, its CA1 and CB1 inputs are used to interface with the UBC as the ODH and CDH signal lines. In addition, the CA2 output is used as the ODACP return handshake line.

8.2.1.1 Incoming Data Handshake. Active transitions on the CA1 or CB1 signal lines set bits 1 and 4 respectively in the PORT 1 interrupt flag register (IFR). Testing of these bits by the code in the GETCH section of the routine allows for the proper data ready indications. If neither bit is set, the code flow falls through to the timer section to test for word interval timeouts. If bit 1 is set, the code transfers control to the data retrieval section, GETDAT. If bit 4 is set, the code transfers to the command data section to



IC-56860

Figure 8.2-1: Harris BIU INDEV Subroutine Flow Chart



IC-56859

Figure 8.2-1: (Concluded)

allow for a handshake response without further processing. This is necessary since the Graphic 7 ignores command data words from the Harris.

8.2.1.2 Return Handshake. Whenever data is read from PORT 1A via a load command, the VIA will set the CA2 signal line according to the state of the PCR initialization. In the case of the Harris BIU, the P1PCR was set to respond with a 550 nsec positive pulse (+5 V TTL) as the return handshake. It should be noted that this pulse is not a true mimic of the standard Graphic 7 interface in which level handshaking is used. However, this pulsed handshaking was necessary to insure that the signal level on the ODACP line would return to zero after the Harris UBC dropped the incoming data handshake signal and no erroneous handshake would be detected from the special hardware configuration used (See Paragraph 5.2.2).

8.2.2 Command Data Processing

As mentioned above, the Graphic 7 ignores command data from the Harris UBC. To allow for proper handshake processing, the BIU must insure that there are no previous data words waiting to be transmitted and then start the DMA timer as if the first word of a new DMA operation has just been received. This requirement has led to the special CDH processing in the GETCDA section. The first check in this section of the code is to determine if it is necessary to send a packet with previously received data. The INDONE section of the code is used to accomplish this, if necessary. If no operation was previously in process, the data word is treated as the start of a new DMA operation and the associated timers are set. Prior to entering the CHARS (word processing) section of the code, a check of the type of data being processed causes a return for the next character if a command data word was indicated. This halts the transmission of the unneeded command data word on the network.

8.2.3 Interrupt Processing

Unlike the Graphic 7 BIU, the Harris BIU must generate interrupts of the Harris host. To support the Harris UBC handler, both input and output interrupts are necessary. In addition, a status line used to indicate which type of inter-

rupt occurred is controlled by the BIU software. This interrupt processing is controlled by the $\overline{R}/W$ and IIFU manual handshake lines.

8.2.3.1 $\overline{R}/W$ Handshake Line. The $\overline{R}/W$ status line is controlled by the output of bit 5 of the PORT3A. With a high level written to PORT3A in this bit location, the $\overline{R}/W$ signal line goes to the low state because of the hardware inverter. Likewise, by writing into PORT3A with this bit set low, a high level is present on the $\overline{R}/W$ signal line. As was mentioned above, the line being high signifies that the last IIFU signal indicated a successful write operation from the Harris UBC. For this reason, the Harris BIU responds to a word transfer in INDEV by writing into PORT3A the word HARADW.

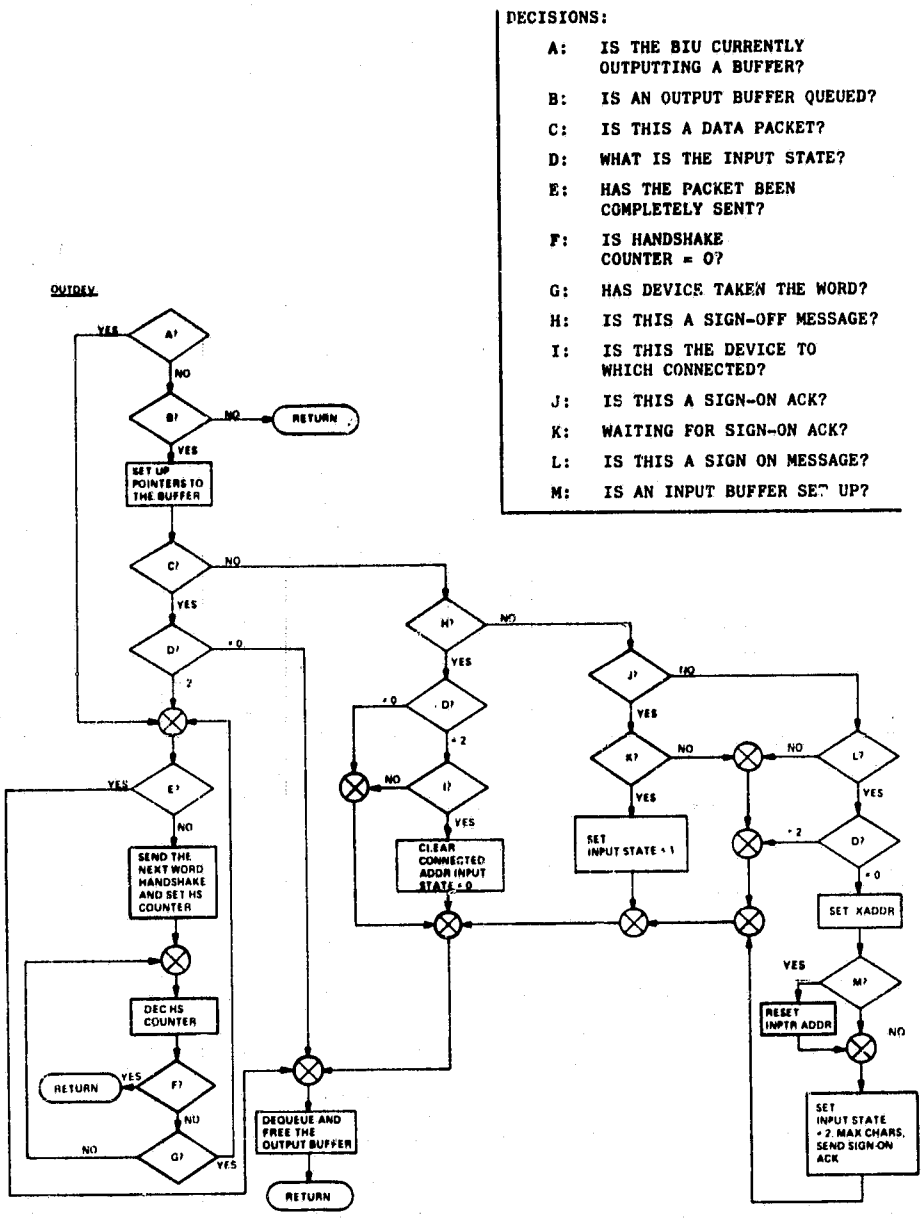
8.2.3.2 IIFU Handshake Line. To insure that the $\overline{R}/W$ signal line is interpreted properly, the IIFU handshake is sent immediately after the state of the $\overline{R}/W$ line has been set. In the INDEV subroutine the IIFU signal is sent by a manual handshake operation caused by the storing of the word IIFU in the P3PCR. The output of the CB2 signal on the PORT3 VIA is set to a high level by this word. The signal is held high until cleared by storing the value NIIFU in P3PCR. This is accomplished after the UBC has had a chance to process the interrupt and is controlled by a counter loop which takes approximately 10 usecs to execute.

8.3 OUTDEV Subroutine

As in the case of the INDEV Subroutine, OUTDEV must perform all of the necessary Graphic 7 software functions which control data input to the Harris UBC. The functions which are unique to the Harris BIU include the handshake processing, interrupt control, deadlock control, and the handling of sign-on requests. These functions are described in the paragraphs below and the flow chart for this routine appears in Figure 8.3-1.

8.3.1 Output Handshaking

As with input handshaking the output handshaking process is designed to insure that data words cannot be lost during transfer. The fully handshaken interface described in Sec-



ORIGINAL PAGE IS
OF POOR QUALITY

IC-56868

Figure 8.3-1: Harris BIU OUTDEV Subroutine Flow Chart

tion V is implemented using the software constant DATU and an interrupt control of the IIFU signal line similar to that in the input sequence.

8.3.1.1 Output Data Valid. Here again, the storing of a data byte into the output port, PORT 2A, triggers an automatic handshaking signal by setting the CA2 output to a high state. This signal is cleared by the active transition of the CA1 signal line which indicates the Harris UBC has accepted the data word.

8.3.1.2 Output Return Handshake. As in the case of the input return handshake, the Harris signals that the data sent by the BIU has been processed by raising the DATU signal line. The active transition of the signal from low to high on the CA1 signal line, is used by the 6522 to clear the CA2 output and set the interrupt flag bit in the IFR. The software uses a wait loop to repeatedly check this bit location to determine if the data has been processed.

8.3.2 Interrupt Control

As was mentioned in Paragraph 8.2.3, the interrupt signal line to the Harris CPU interrupt logic, IIFU, is used to signal when the connected device has processed an output from the UBC as well as when the device has an input word for the UBC. In this case an input word has been placed on the input bus and the Harris CPU must be notified to read the word for processing. Here again, a manual handshake is used by the BIU. The $\overline{R}/W$ signal line must also be set to reflect the proper state.

8.3.2.1 $\overline{R}/W$ Signal Control. The $\overline{R}/W$ signal line is set to reflect the meaning of the last IIFU signal sent to the UBC. The state of this line must be set before the IIFU signal is sent. This is necessitated by the delay involving the software control of the signal line in comparison to the speed of the hardware processing in the UBC. The data word HARADR is stored into the PORT3A register, thus setting the output on data line 5 to a high state. The 6522's data line is inverted by the hardware to indicate to the UBC that the IIFU signal is the result of a read operation.

8.3.2.2 IIFU Signal Control. The state of the IIFU signal line is controlled by a manual handshake using the CB2 output of the PORT3 VIA. The value IIFU is stored into the PORT3 PCR which causes the low-to-high transition of the CB2 signal line which is connected to the IIFU data line to the Harris UBC. In this case, however, the signal is held until the interface has acknowledged the receipt of the data on the input bus. At that point the NIIFU data word is stored into P3PCR, lowering the interrupt signal.

8.3.3 Deadlock Control

The possiblility of deadlock on output from the BIU arises since the BIU enters a tight loop monitioring the state of the CA1 signal line on PORT2 to observe the transition from low to high. The BIU does not use interrupts from the UBC to indicate when data is available on the PORT1 bus, so it is possible for the UBC to be trying to transmit data to the BIU at the same time the BIU is transmitting data to the UBC. In this situation, both devices would raise their valid output data signals and wait for the return response before proceeding. To prevent this deadly embrace, the BIU must defer to the UBC and respond to its output. To accomplish this, the "X" register is loaded with the value $E0 and then decremented on each pass through the wait loop. If the "X" register reaches zero before the return handshake is received (approximately 2.0 msecs) the output is terminated and the OUTDEV routine is exited. Since the pointer to the next byte to be transmitted is still in the variable OUTBC, no data is lost and the OUTDEV subroutine will resume transmission of data upon return during the normal MAIN LOOP calling sequence. Since control is returned to the MAIN LOOP, the INDEV routine is called before return to the OUTDEV routine, thus enabling the clearing of the deadlock situation.

8.3.4 Sign-On Acknowledgment Processing

The Harris BIU is designed to attempt a connection with the backboard BIU whenever possible. This is to allow the operation of the VOLLEY diagnostic program. Since the backboard is the only BIU to which the Harris BIU may connect, the sign-on acknowledgment section of the OUTDEV routine is special. In this section, the CONECT flag is set to zero to indicate a connection with the backboard and the WAIT flag, used to indicate that the BIU is awaiting the response to a sign-on request, is cleared.

8.3.5 Sign-On Request

The requirement to process outputs from the VOLLEY diagnostic program dictates a modification to the normal sign-on request processing. Since the RESET routine establishes a link with the backboard, the INTBUF routine allocates the first buffer with the backboard's address set in XADDR and the TO address of the packet. The Harris BIU must insure that a sign-on request from a Graphic 7 BIU supersedes the existing backboard connection since the Graphic 7 system has priority over the diagnostic process. To insure that the preallocated packet which was set aside by INTBUF, does not result in lost data, the sign-on processing section of OUTDEV must reset the values in XADDR and the third byte of the allocated packet. If a packet was not preallocated (INSET = 0) the TO address does not require resetting and this section of the code is bypassed. The CONECT flag is set to 1 to indicate a connection with a terminal device and control passed to the packet freeing section of the routine.

8.4 CKTOUT Subroutine

In the Harris BIU software, only two timers are used. These are a timer set when the BIU is attempting to connect with the backboard and the standard status message timer. These two functions are described below and the subroutine flow chart appears in Figure 8.4-1.

8.4.1 Sign-On Acknowledgment Timer

If the attempt to connect with the backboard BIU fails, the value in TOD exceeds the value stored in TSACK during a call to STIMER. This results in the WAIT and XADDR flags being cleared and the CONECT flag being reset to -1.

8.4.2 Status Packet Timer

The value in TSTAT is used to determine when the next status packet should be sent. When the TOD variable exceeds this value the control of the CKTOUT routine passes to the SNDSTA subroutine.

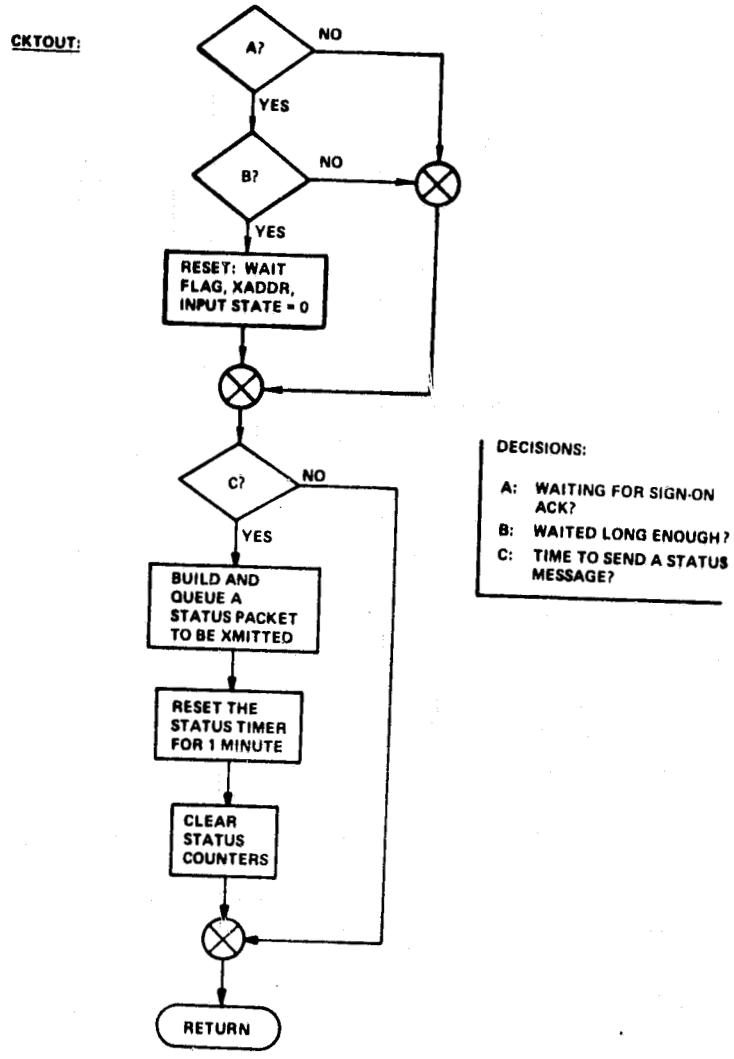


Figure 8.4-1: Harris BIU CKTOUT Subroutine Flow Chart

8.5 IRQ Routine

As was mentioned in Paragraph 7.11, the interrupt handling routine in the Harris BIU must recognize two types of interrupts: the standard network receiver interrupt and the special UBC DISC interrupt. This second interrupt is required to insure that the Harris CPU has a means of resetting the device interface during power-up or reboot operations. The DISC signal line is connected to the CA1 input on the PORT3 VIA. The PORT3 IER is initialized during the RESET routine to trigger a IRQ whenever there is an active transition on the CA1 control line. This transition forces the BIU's CPU to transfer control to the IRQ Routine via the hardware vector located at address $FFFE. Upon entering the IRQ routine, a test is made of the P3IFR register to determine if the interrupt was the result of the DISC signal. If so, the PORT3A data lines are reset to allow input from the UBC's UR register. The value on the UR lines is compared to the valid address for a Graphic 7 ($0F) and, if found to be correct, the byte is loaded from PORT3A triggering the return CNCT signal from CA2. The PORT3A data lines are then returned to the output state to insure the $\overline{R}/W$ control line can be used and control passed to the interrupt return section of the routine. If the UR register does not match, the data is ignored and no return handshake is issued. Figure 8.5-1 is provided to detail the difference between the standard IRQ routine and the special Harris IRQ routine.

8.6 Constants

The constants section of the BIU code contains the BIU description sent with all status packets. In the case of the Harris BIU, these 12 bytes form the message "HARRIS BIU ".

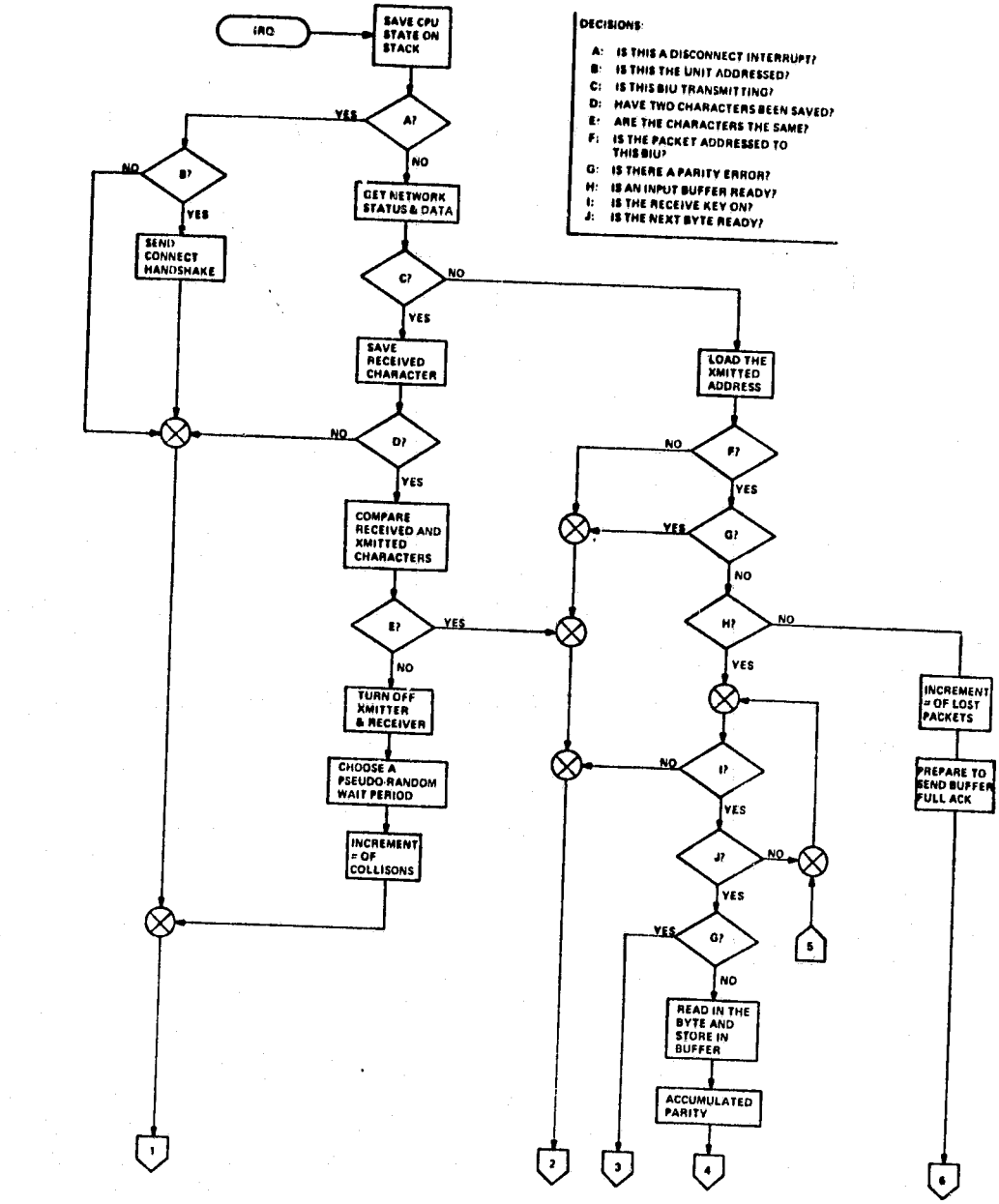


Figure 8.5-1: Harris BIU IRQ Routine Flow Chart

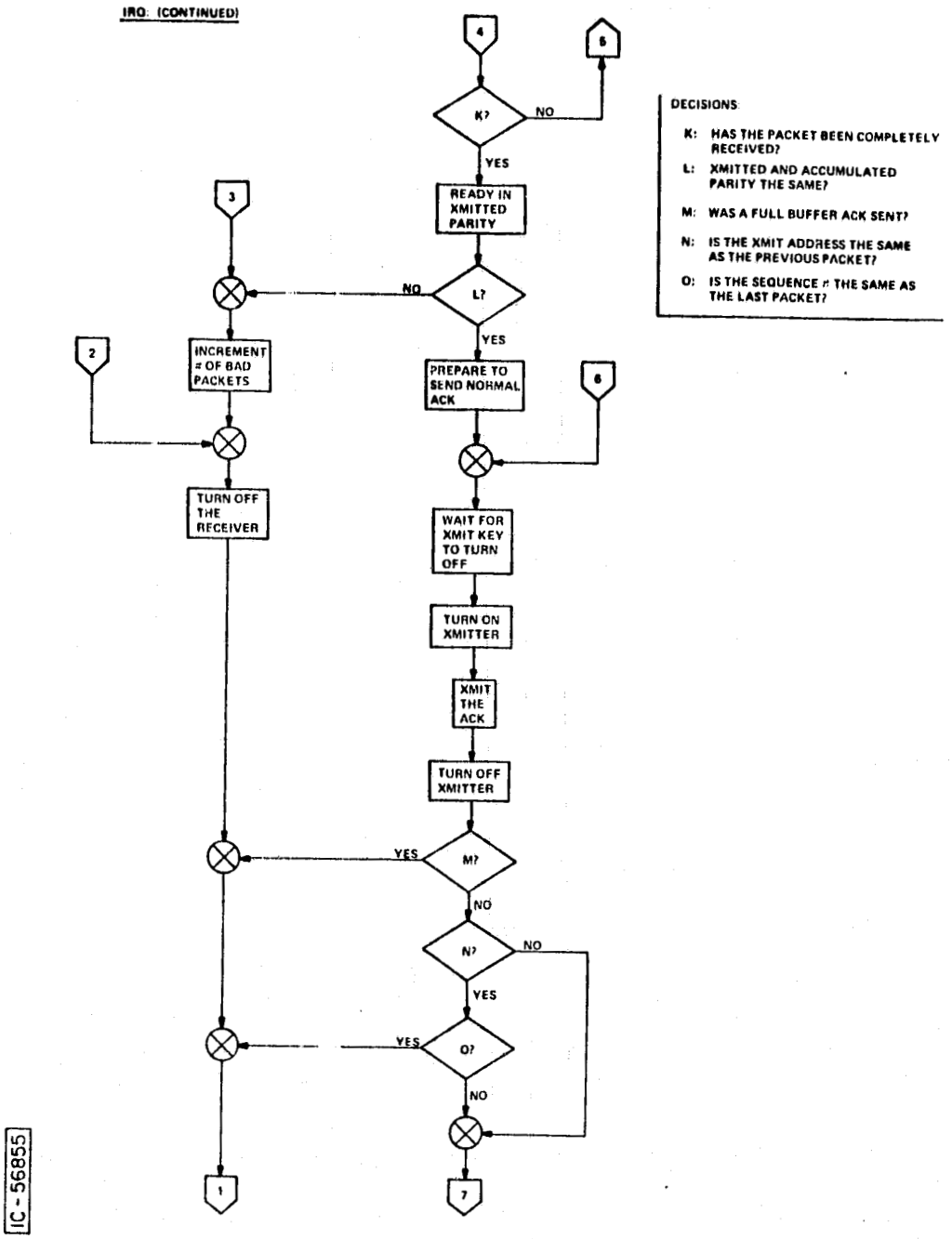


Figure 8.5-1: (Continued)

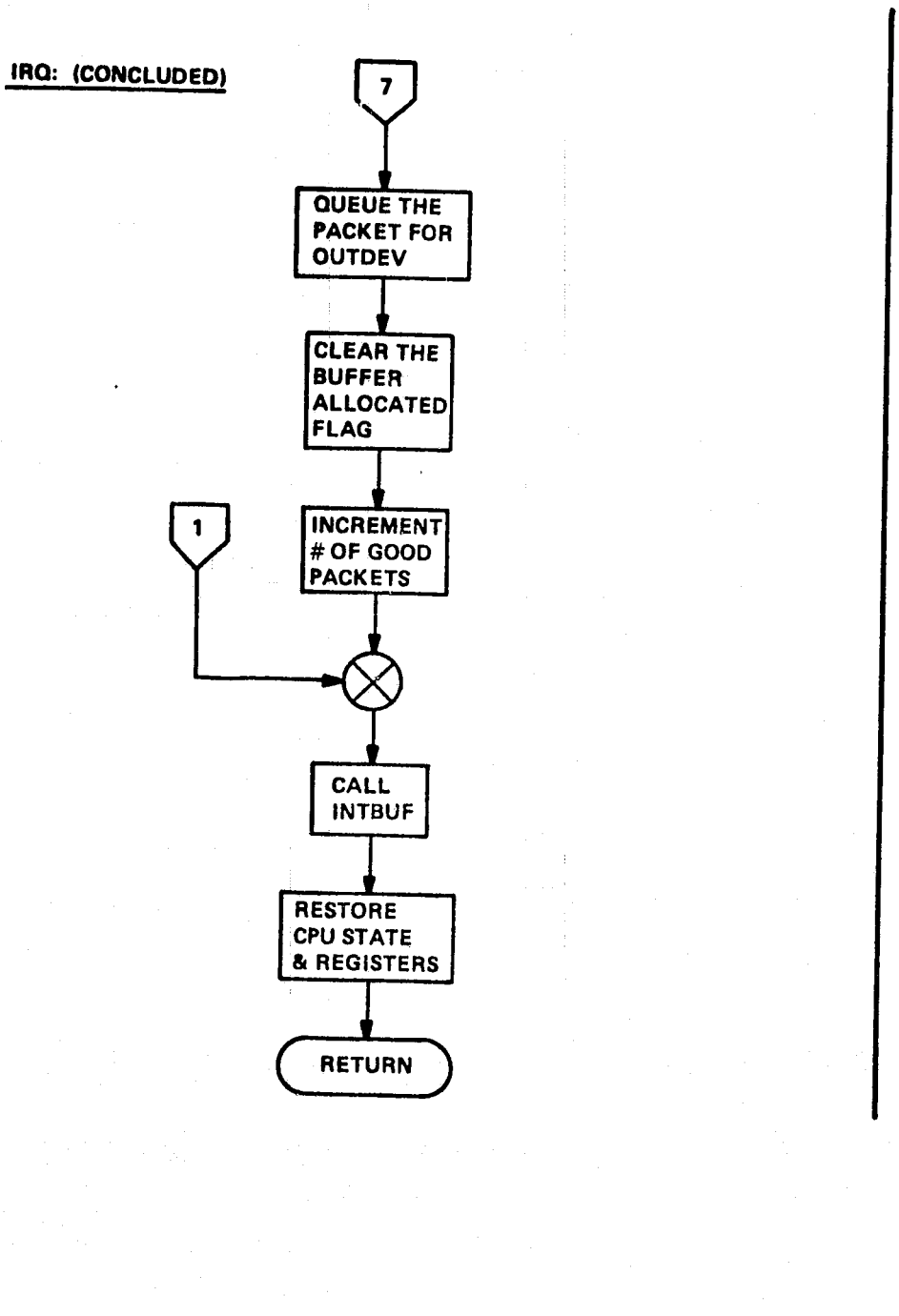


Figure 8.5-1: (Concluded)

SECTION IX

SPECIAL GRAPHIC 7 BIU SOFTWARE

9.0 INTRODUCTION

The main purpose of the special Graphic 7 software is to cause the BIU to appear as the Harris host computer to the Graphic 7 parallel interface. Additionally, logical connection to a Graphic 7 BIU or any other network subscriber is supported. These interconnections are handled through the INDEV subroutine.

9.1 RESET Routine

As in the case of the Harris BIU, several areas of the RESET routine are unique to the Graphic 7 BIU. These include the handshaking initialization, CNCT/DISC cycle, power-up cycle and the special "WHICH SYSTEM?" request. These areas are detailed in the following paragraphs and the flow chart for the Graphic 7 RESET routine appears in Figure 9.1-1.

9.1.1 Handshake Initialization

To establish the all-important handshaking logic control of the parallel interface, the 6522 VIAs must be initialized to respond to the appropriate signal level changes on the interface signal lines. In the RESET routine of the Graphic 7 BIU, the VIAs' Peripheral Control Registers (PCRs) are set according to the values in Table X below. As in the Harris BIU, PORT1 is established for input from the device and to recognize the low-to-high transition of the CA1 signal line as indicating that valid data is present on the input data bus. PORT2 is set as the output bus. It uses the CA2 signal line as the valid data signal line and recogizes the CA1 signal line as the device acknowledgment of the data on the output bus. Once again PORT3 is used for some special functions, such as establishing the CNCT/DISC signals using CA1 and CA2, and reading in the BIU's home address on PORT3B. The use of the interrupt capability of the VIA is not necessary in the Graphic 7 BIU since all inputs and outputs are fully handshaken and no special interrupts are generated by the Graphic 7.

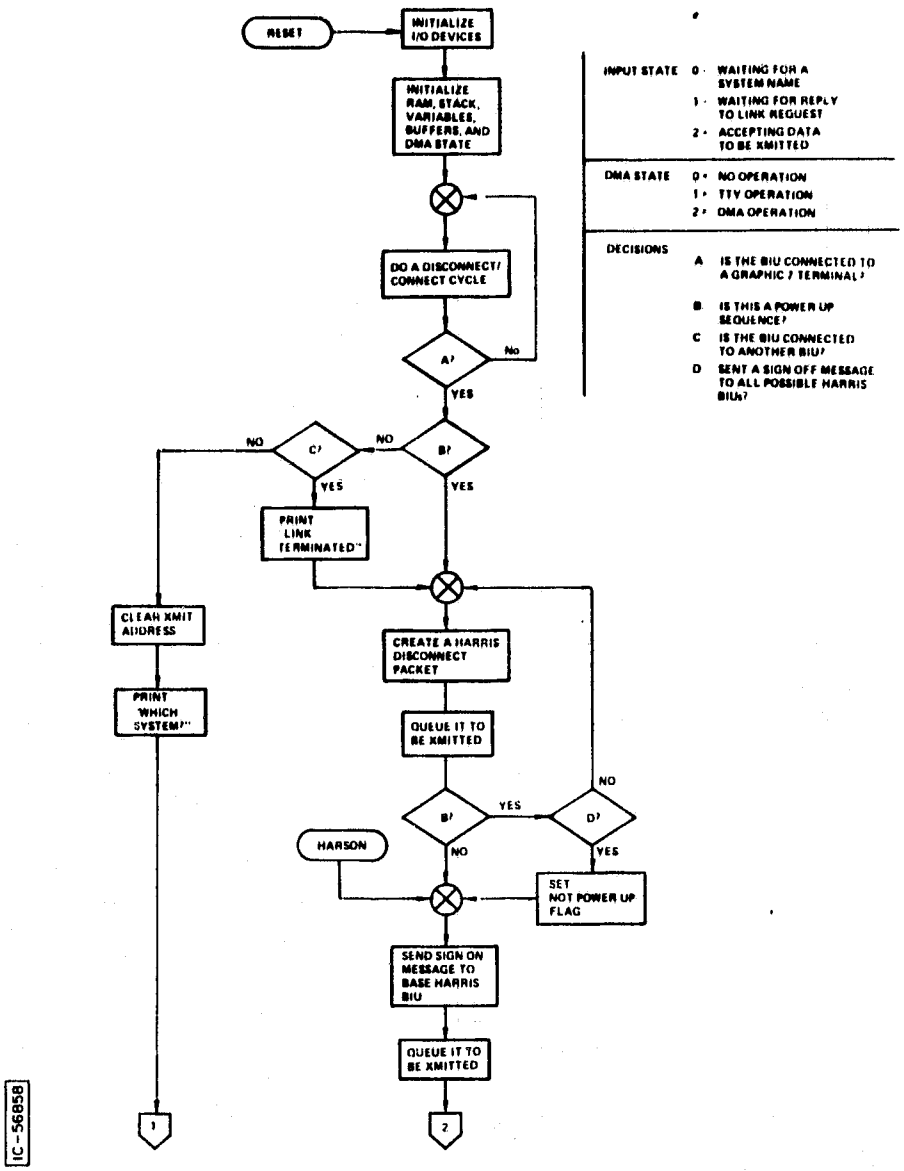


Figure 9.1-1: Graphic 7 RESET Routine Flow Chart

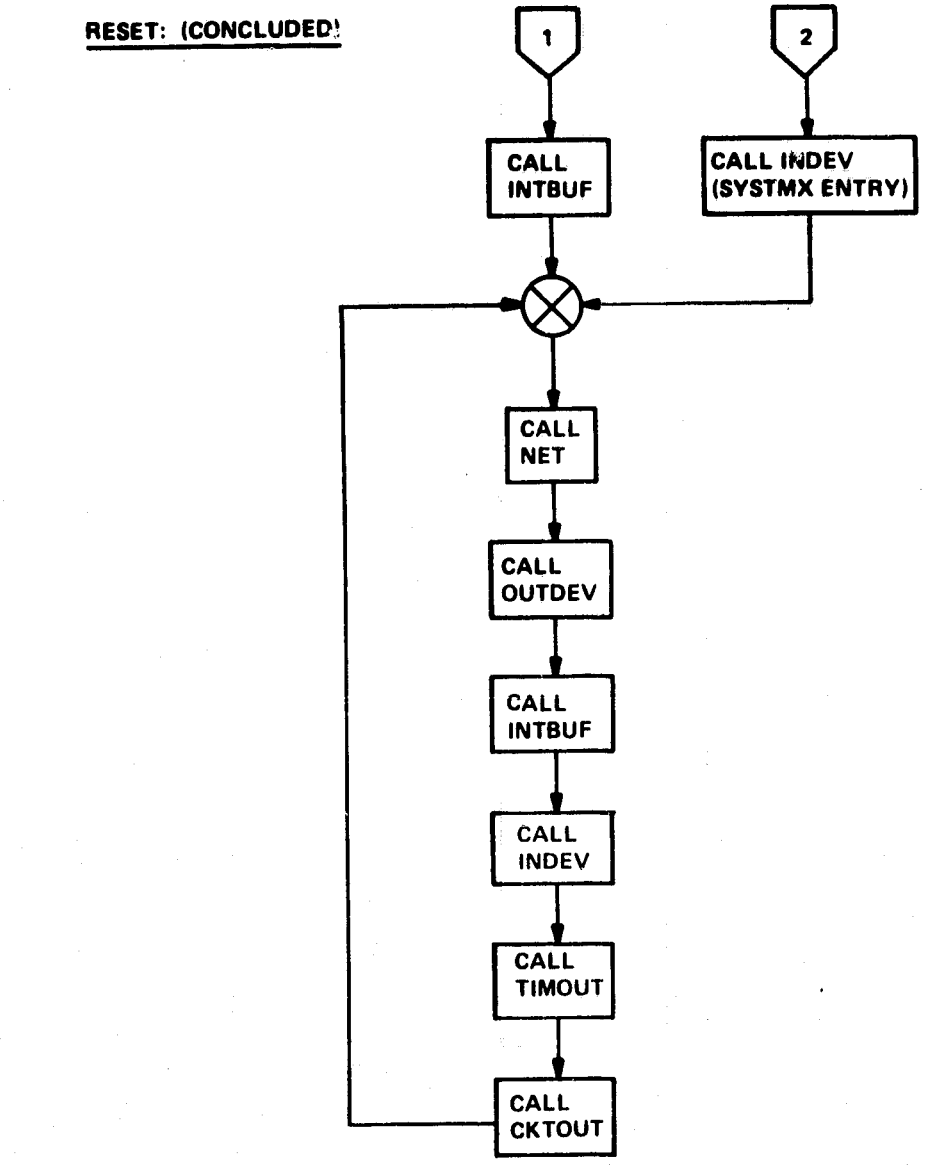


Figure 9.1-1: (Concluded)

TABLE X

GRAPHIC 7 BIU HANDSHAKING CONTROL

PORT #	DEFINITION	INIT CODE	ACTIVE TRANSITION CA1	CB1	RESPONSE CA2	CB2
1	INPUT	10011011	L-TO-H	N/A	PUL	N/A
2	OUTPUT	10001000	H-TO-L	N/A	LEV	N/A
3	CONTROL	11101010	L-TO-H	N/A	PUL	MAN

L-TO-H = Low-to-High (0V -to- +5V)
H-TO-L = High-to-Low (+5v -to- 0V)
PUL = 550 ns +5V Pulse
LEV = +5V Level held until active transition on CA1
MAN = Manual Handshake under software control
N/A = Not Used

9.1.2 CNCT/DISC Cycle

To allow for the accurate mimicking of the Harris UBC, the CNCT/DISC cycle provides the connection logic used by the Harris to control the daisy-chained bus. When power is applied to the BIU or whenever the RESET button is pressed, the DOCNCT section of the RESET routine transmits a 550 nsec pulse on the CA2 signal line of PORT3. This pulse is a direct result of the store into PORT3A which activates the pulsed handshake of the PORT3 VIA. In addition to triggering the DISC signal, the unit code for the Graphic 7, which is presented on the low-order 4 bits of PORT3A, is set to zero by the data stored in the VIA. Note that the digital hardware of the BIU inverts the data from the low-order 4 bits for this purpose. The tri-state control for the unit code bus is also set by this word since the control bit, bit 4, is set to a high state thus indicating the low-order 4 bits are for output from the BIU (See Paragraph 5.2.5). The code loops at this point until the interrogation of P3IFR indicates that the active transition of the CNCT signal has been received on CA1.

9.1.3 Power-Up Cycle

As in the Harris BIU, constraints on user transparency require that the power-up or power-fail condition in the

Graphic 7 BIU forces an attempt to connect, when possible, to any available Harris BIU without user interaction. This necessitates a complex power-up sequence in the Graphic 7 BIU. With multiple terminals connected via the bus, the virtual circuits created by the BIU interconnections are destroyed by a power failure or removal of power (turning the BIU off) at either end. To allow for a smooth reconnection of these circuits when power is restored, each BIU executing the RESET code issues a sign-off message to the other BIU comprising the old circuit. This is to prevent the possible loss of resources if one BIU assumed it was still in the connected state (See Paragraph 7.5.3.2). Since the previous state of the BIU is lost when power is removed (dynamic RAM), the BIU sends a sign-off message to all possible Harris BIUs when executing the RESET code during a power-up sequence. This is not the case, however, when the RESET button is pushed. In this case, only one sign-off message is necessary and that is to the BIU whose address is still present in XADDR. In the case of a power-up cycle, the Graphic 7 BIU attempts to connect to any available Harris BIU after the transmission of the string of sign-off messages. Since there is normally one Harris BIU for every Graphic 7 BIU in the system there is a high probability of finding at least one available BIU for the completion of the virtual circuit. The criterion used to distinguish between the pressing of the RESET button and a power-up cycle is based on the loss of data due to the dynamic RAM. If the variable HARGO contains the bit pattern $57 then the code assumes this is not a power-up cycle and continues executing the code as if the RESET button was pressed.

9.1.3.1 Sign-Off Loop. The constants HARBAS and TRMCNT are used to establish the limits for the sign-off loop. HARBAS is used to indicate the base network address of all Harris BIUs. It should be the lowest address for any set of Harris BIUs and all other Harris BIUs should have an address which falls within the range between HARBAS and HARBAS+TRMCNT-1. TRMCNT is used to indicate the total number of possible Graphic 7 terminals in the system. In the current application, HARBAS is set to $21 and TRMCNT is set to $08. The sign-off loop repeatedly calls PCONST with a new value of XADDR within the range described above and then calls ENQ with the newly defined sign-off packet. The final result of this loop is a queue of sign-off messages, one to each of the possible Harris BIU addresses in the system.

9.1.3.2 Sign-On Message. Following the generation of the sign-off messages in the power-up cycle, the Graphic 7 BIU attempts to establish a connection with one of the Harris BIUs by sending a sign-on request packet. To aid in reducing network overhead, a special algorithm is used to establish to which Harris BIU the first connection message is sent. The algorithm uses the low-order 4 bits of the Graphic 7 BIU's home address and the high-order 4 bits of the Harris base address to form the 8-bit address byte for the sign-on request packet. If that Harris BIU is unavailable for the connection, the address is incremented and tried again (See Paragraph 9.4.2). It should be noted that no sign-on message is generated when the user presses the RESET but a dialogue with the terminal user is initiated (See Paragraph 9.2.4).

9.1.4 "WHICH SYSTEM?" Request

If the RESET button is pressed or no Harris BIU is available for a connection, the RESET routine engages in a dialogue with the terminal user to establish a connection with some other element on the network. To accomplish this, the Graphic 7 BIU sends, if necessary, a sign-off message to the previously connected BIU and a "LINK ENDED." message to the terminal. Next, the BIU sends the "WHICH SYSTEM?" message to the terminal via the PUTSTR subroutine. As in the case of the "LINK ENDED." message, this message is located in the high end of the EPROM memory and indexed based on the address of the variable ASCII. The "Y" register is used as the index for this reference. The response to this interrogation is processed via the INDEV routine (See Paragraph 9.2.4).

9.2 INDEV Subroutine

As in the case of the Harris BIU, the INDEV routine is the main input interface with the attached device and as such must insure that all handshaking and data transfer operations are performed in accordance with the standard interface protocol. In addition, to provide for connections to other BIUs, the Graphic 7 version of INDEV must process the "WHICH SYSTEM?" response from the device as well as requests for a connection from other terminals on the network. The CONECT flag is used to distinguish between these three input states. Also, there are two modes of data input from the Graphic 7 which must be distinguished: DMA and Teletype. All of those functions and the other major areas of difference between the

Graphic 7 version of INDEV and the Harris version are detailed in the paragraphs below. Figure 9.2-1 is the Graphic 7 INDEV flow chart.

9.2.1 DMA/TTY Operations

The two modes of operation for the interface cannot be distinguished. In TTY mode, all data words contain a null byte followed by valid data in bit locations 7 through 0. This valid data byte must be examined for control information, such as carriage return, line feed or rubout characters. It is unwise to try to determine the mode of operation by examining the characters that are being transmitted since the DMA mode has no restriction on the data in either byte of the word. For this reason, two timers are used to establish the mode of any given input operation.

9.2.1.1 DMA Timer. The DMA timer is more accurately described as a counter. When the first input word of an operation is detected by the handshake signal, the value of $20 is stored into the variable location DMATM. The input word is processed and the control returns to the handshake test part of the routine where the counter is decremented until either a new input valid handshake signal is detected or the counter reaches zero (approximately 650 usecs). If a new handshake is detected, the timer is reset, the data processed and the sequence repeated. When the counter reaches zero, the end of the DMA operation is indicated by setting the DMATTY flag to zero. Control is passed to the TTY handler section of INDEV. If more than one word is processed without DMATM reaching zero ("Y" register > 11), it is assumed that a DMA operation has occurred and the buffered data sent without delay. If only one word is in the buffer, the word is probably a TTY word and the low-order byte is examined for control characters. If any control characters are found, the buffered packet is transmitted and control returned to the MAIN LOOP. If no control characters are found, control is passed directly to the MAIN LOOP to await additional characters or the expiration of the TTY timer.

9.2.1.2 TTY Timer. The processing scheme described above would lead to the unnecessary and perhaps damaging delay of one-word DMA transactions if no method of clearing these one word packets existed. For this reason a TTY timer is used to trigger the sending of data packets that are

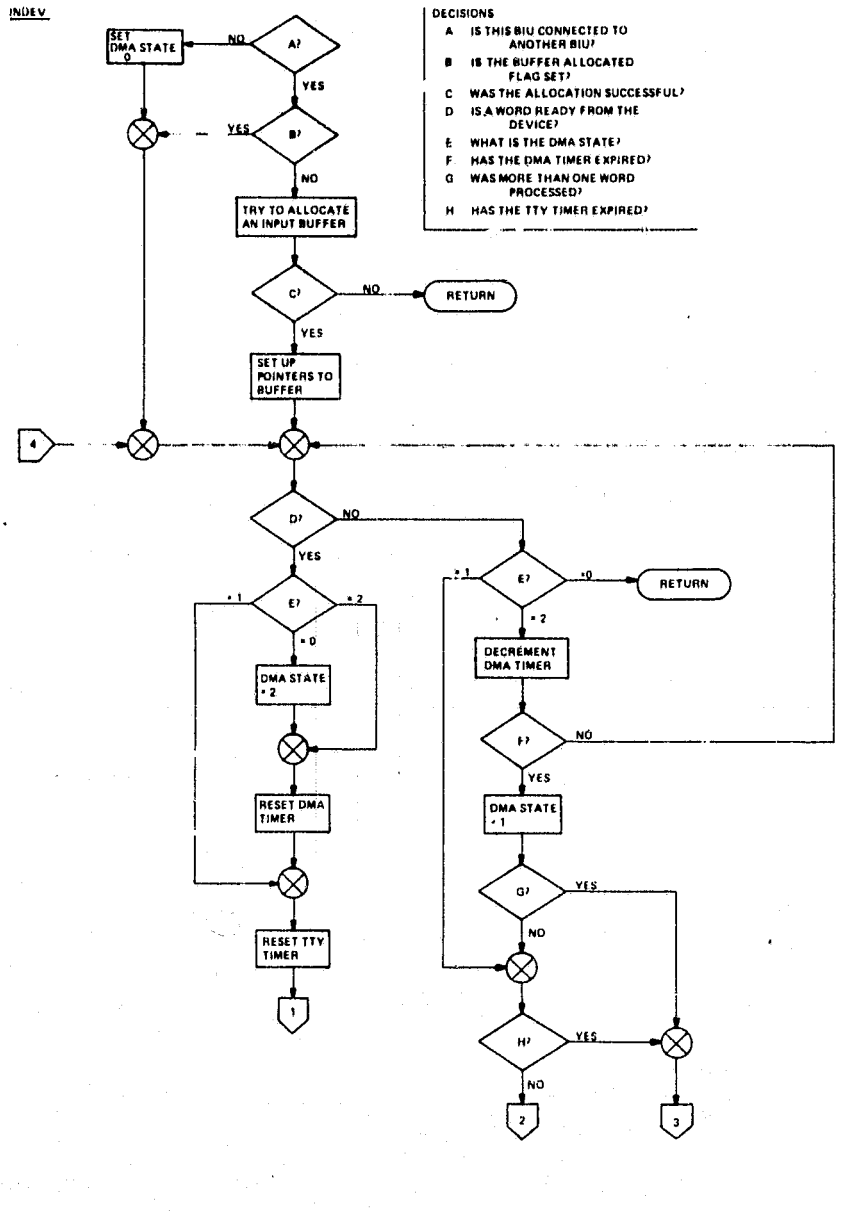
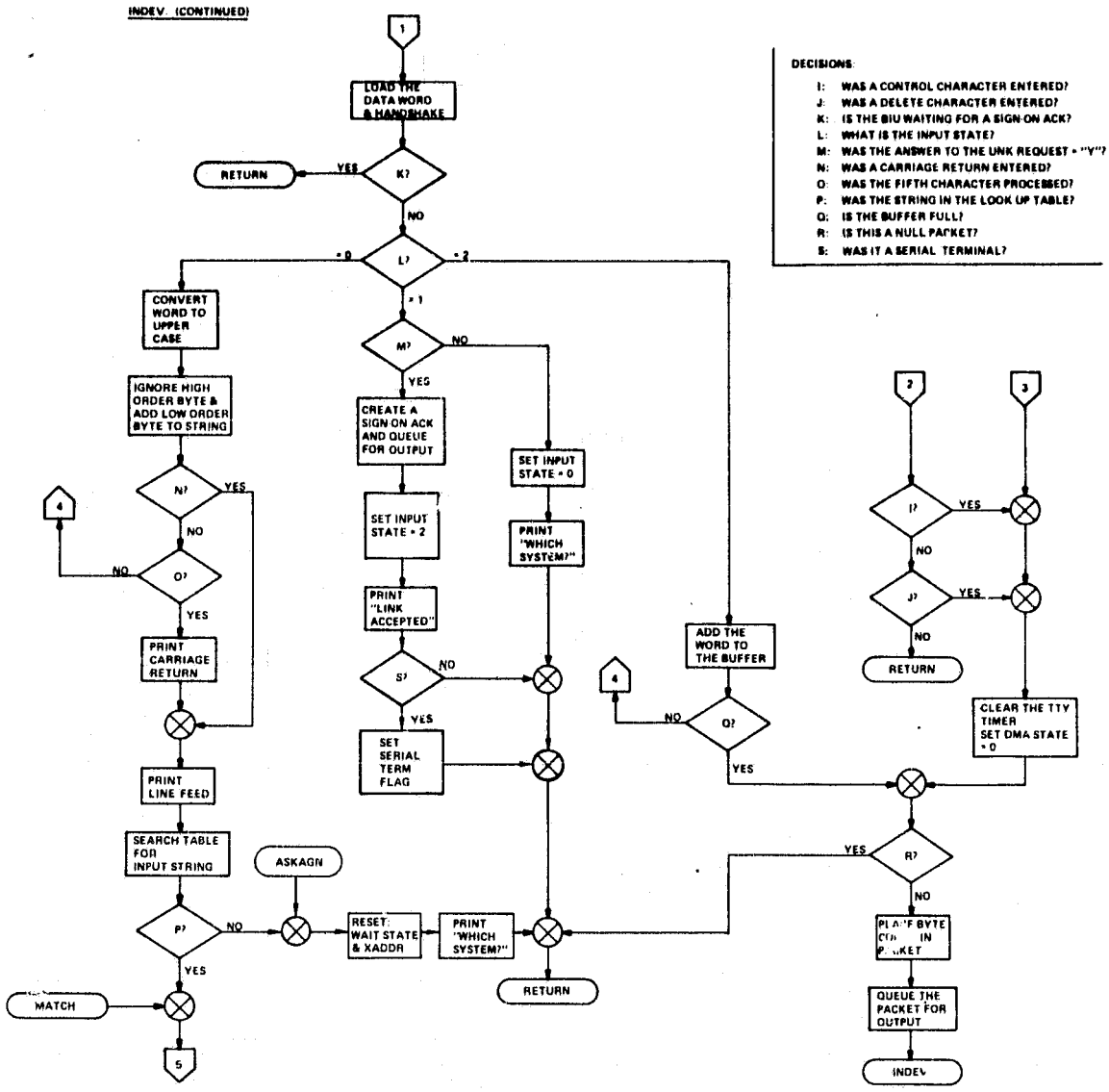


Figure 9.2-1: Graphic 7 BIU INDEV Subroutine Flow Chart



IC-56865

Figure 9.2-1: (Continued)

INDEV: (CONCLUDED)

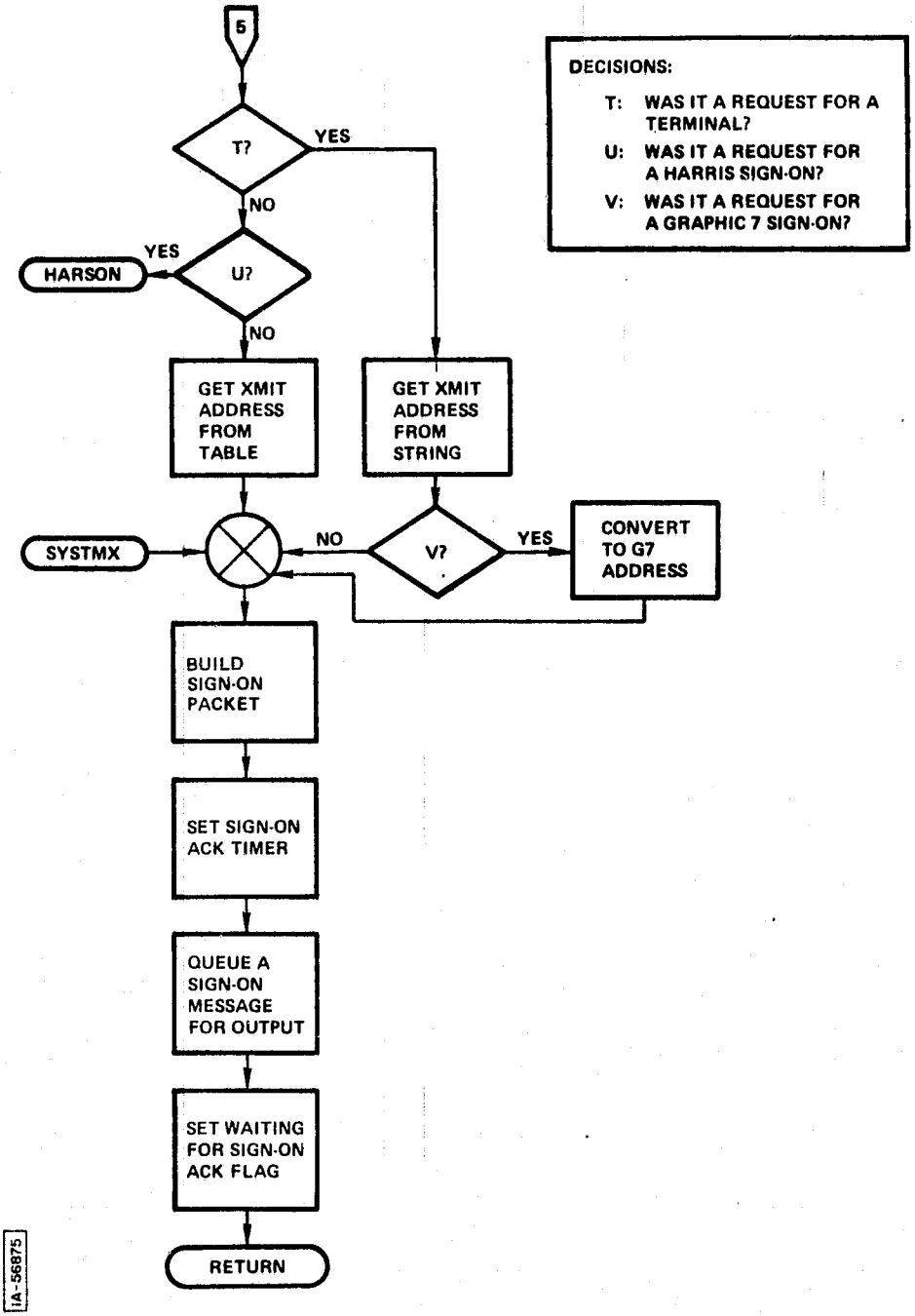


Figure 9.2-1: (Concluded)

buffered and awaiting the receipt of a control character. The variable TTYTM is set as a flag whenever a word is processed by the interface. The STIMER routine is used to mark the absolute system time when this flag should be cleared (See Paragraph 7.14). STIMER uses the value stored in the "A" register as an increment to the TOD clock for storage in the 3-byte variable indexed by the "X" register. In this case the variable set is TETTY. The CKTOUT subroutine, when called by the MAIN LOOP, examines this variable to determine if the TOD clock has exceeded it (See Paragraph 9.4.3). For the purpose of the TTY timer, if the TOD clock exceeds the TETTY variable value, TTYTM is cleared to indicate the expiration of the TTY timer. The value $4 loaded into the "A" register prior to the call to the STIMER routine results in a one second delay before the clearing of the TTY timer flag. When the TTYTM flag is cleared, the next pass through INDEV results in the transmission of any buffered data by transferring control to the INDONE section of the code.

9.2.2 Graphic 7 Handshaking

As in the case of the Harris BIU handshaking signal lines, the Graphic 7 signal lines must respond in a manner similar to the Harris UBC during any given data transfer. Once again, the PORT 1 VIA is used for input to the BIU and the CA1 input to this VIA is used to indicate that valid data is ready on the input bus. The data acknowledgment handshake is transferred automatically by the VIA on the CA2 output. Unlike the Harris BIU, however, the secondary input handshake on CB1 is not used.

9.2.2.1 Incoming Data Handshake. The hardware on the Graphic 7 parallel interface card generates both the IIFU signal and the IWR signal when valid data is ready on the ID00-15 bus. The Graphic 7 BIU ignores the IIFU signal since input is not interrupt driven. The low-to-high transition of the IWR signal, however, drives the PORT 1 CA1 input and indicates that data is ready on the PA0 through PB7 signal lines. This transition of the IWR signal sets bit 1 of P1IFR thus enabling the test of this bit location on a polled basis in the INDEV routine.

9.2.2.2 Return Handshake. Whenever data is read from PORT 1A via a load command, the VIA sets the CA2 signal line

according to the state of the PCR initialization. In the case of the Graphic 7 BIU, the PlPCR is set to respond with a 550 nsec positive pulse (+5 V TTL) as the return handshake. It should be noted that like the Harris BIU this pulse is not a true representation of the standard interface since level handshaking is used. However, this pulsed handshaking is necessary to insure that the signal level on the ICTL line returns to zero after the Graphic 7 drops the incoming data handshake signal (See Paragraph 5.2.2).

9.2.3 Terminal Input Data Processing

After the data word from the device is loaded and the handshaking sequence completed, the BIU must determine which of the three processing options to pursue. The CONECT flag is tested to determine if the incoming data is in response to the "WHICH SYSTEM?" interrogation, in response to a sign-on request from another terminal or destined to be transmitted on the network as part of a data packet.

9.2.3.1 Network Data processing. When the data word is destined for the network (CONECT = 1), the word is placed in the outgoing packet by the CHARS section of the routine. In this case the data packet was reserved by a previous call to the INTBUF subroutine. The words are stored in a buffer indicated by INPTR as they arrive until one of the timers described above expires or the maximum number of bytes in a packet (MAXPAX) is reached. In either case, control is passed to the INDONE section of the code to place the packet on the output queue and control returned to the top of the INDEV subroutine.

9.2.3.2 "WHICH SYSTEM?" Response. One of the possible states of the BIU interface is processing input data words in response to the "WHICH SYSTEM?" interrogation (CONECT = -1). After the input word has been acknowledged by the interface handshaking control, the routine jumps to the GETRPY section to determine if the data entered by the user is a valid BIU code name. If the code name is found to be correct, control is transferred to the MATCH section to build a sign-on request message for transmission on the network. At this point several flags are set and control is returned to the MAIN LOOP to await a response.

9.2.3.2.1 Valid System Code. All of the valid system code names that are recognized by a given BIU are stored in the area identified by the variable name TABLE. In the case of the Graphic 7 BIU, each entry in the table is composed of 4 bytes (the 3-byte code name and the one-byte bus address for that device). The GETRPY section of INDEV buffers the words entered by the user until a control character is typed or five words are entered. (Note: In this connection state, data words are assumed to be arriving in the TTY mode and to contain only one valid byte in the low-order byte position). Control is then passed to the code check section where the first three characters of the entry are compared to the table. Codes of five bytes are used in the special case of terminal-to-terminal communication to specify which of the possible terminal addresses is desired. If match is found, control is passed to the MATCH section, otherwise, control is passed to the ASKAGN section of the code where a new "WHICH SYSTEM?" message is generated.

9.2.3.2.2 MATCH Section. This section of the code is used to resolve the special cases involving transmission to a Harris BIU, a terminal BIU or another Graphic 7 BIU. These special cases result in special sign-on sequences to insure that the proper network address is used. In the case of the terminal or Graphic 7 connection request, the network address to be used is based on the value entered in the fifth byte of the interrogation response. In the case of a terminal sign-on request, the fifth byte entered by the user is placed directly into XADDR. In the case of the Graphic 7 sign-on request, however, the high-order 4 bits of the home address are substituted for the high-order 4 bits of the fifth byte and then placed into XADDR. In the case of the Harris BIU sign-on request, the code in the RESET routine to establish the original sign-on request during power-up is used to conserve space in the EPROM. This is accomplished by a jump to HARSON whenever the "Y" register indicates that the Harris code (CAP) was entered. All other valid codes result in a sign-on message based on the address stored in the fourth byte of the table. After XADDR is set, the packet construction subroutine PCONST is called with the sign-on request code stored in the "A" register. When control is returned from PCONST, the ENQ subroutine is used to place the packet on the output queue. The WAIT flag is incremented to indicate that the BIU is waiting for a sign-on acknowledgment packet and control returns to the MAIN LOOP to await a response to the sign-on request.

9.2.3.3 Sign-On Request Response. The third state of the CONECT (0) flag is used to indicate that another BIU has sent a sign-on request packet to this BIU. During the processing of the packet in the OUTDEV subroutine, the CONECT flag was set to zero. At the same time, a sign-on request message was transmitted to the device via the PUTSTR subroutine. The user should respond to this sign-on request packet by entering either a yes or no response on the device. With the CONECT flag set to zero, all entries from the device are treated as a response to this request and thus transferred to the GETYN section of the routine. This routine recognizes only the first character of the word entered, assuming it to be in the TTY format. If the character is a "Y", the code assumes the requested sign-on is to be acknowledged, thus leading to a connection between the two BIUs. Any other response from the device is assumed to be a negative response and results in no response to the requestor, thus leading to a sign-on request timeout (See Paragraph 9.4.2). If the response is affirmative, the PCONST subroutine is used to generate the special sign-on acknowledgment message and the ENQ subroutine is used to place it on the output queue. If the request is from a serial terminal the TERMF flag is set to insure proper processing of data packets in the OUTDEV subroutine (See Paragraph 9.3.1). Finally, the "LINK OKAY." message is set to the user by the PUTSTR subroutine and control returned to the MAIN LOOP for further processing.

9.3 OUTDEV Subroutine

As in the case of the INDEV subroutine, the OUTDEV routine is the primary software interface for the transmission of data from the network to the attached subscriber device. Here again, the interface must perform all handshaking and data transfer functions in a manner similar to the Harris UBC. The OUTDEV routine must also recognize special messages from the network and process data from serial devices in a manner consistant with the parallel interface to the Graphic 7. The special functions to perform these operations are detailed in the paragraphs below. The Graphic 7 OUTDEV flow chart is provided (Figure 9.3-1) for easy reference.

9.3.1 Serial Terminal Output

The Graphic 7 BIU has the capability to communicate directly to serial devices connected to the network through standard serial BIUs. The standard serial BIU data packet

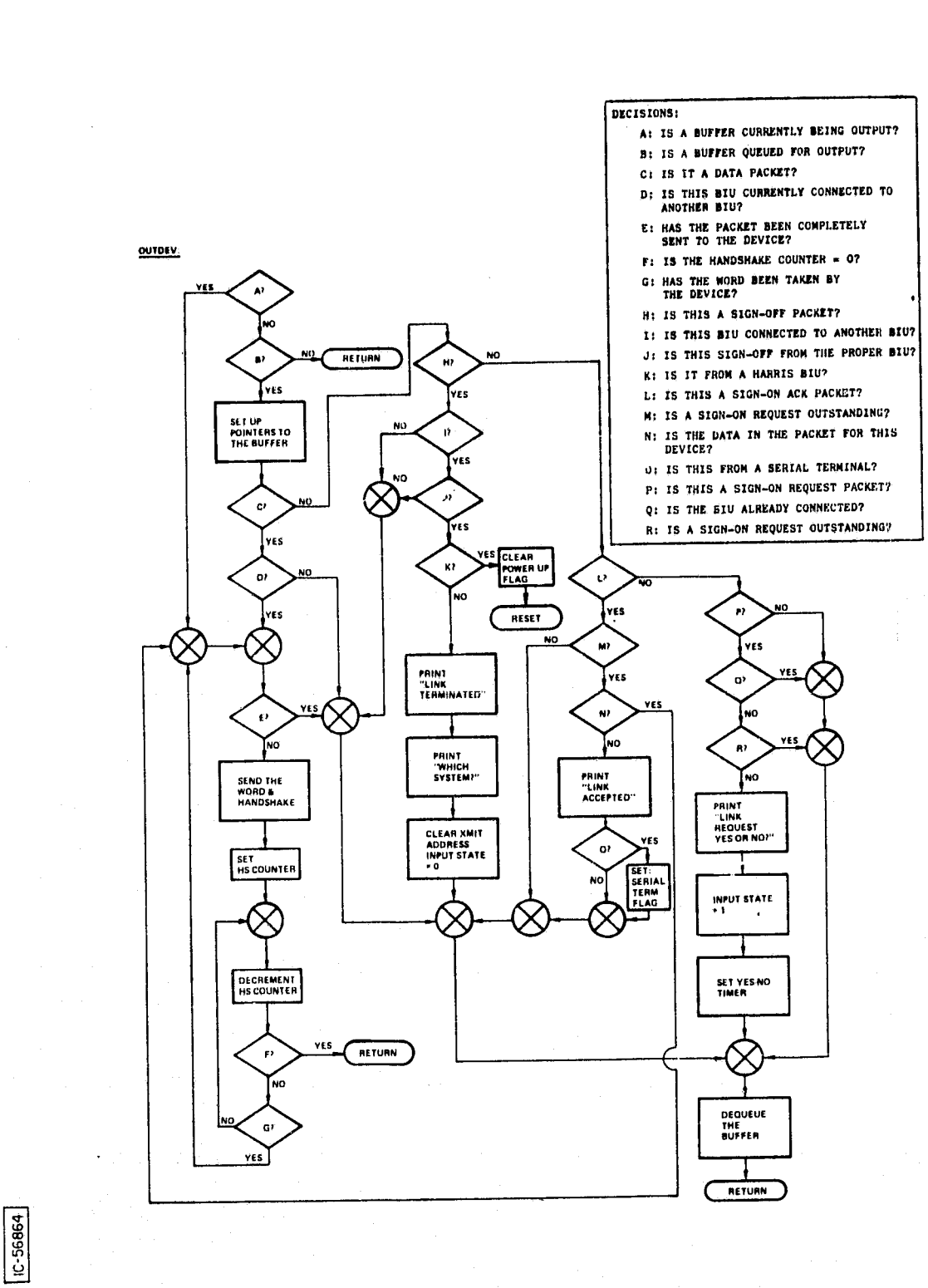


Figure 9.3-1: Graphic 7 BIU OUTDEV Subroutine Flow Chart

contains valid data in each byte of the packet. The Graphic 7 BIU parallel interface operates with a word-at-a-time protocol that transmits two network data bytes at a time to the terminal. When operating in the TTY mode, the terminal ignores the high-order byte of every word thus leading to lost data from serial devices if the standard parallel interface code is used. For this reason, a special handler in the BIU is used whenever the TERMF flag is set indicating a connection with a serial device. This code inserts a null byte in front of every data byte received in a data packet generated by a serial BIU.

9.3.2 Output Handshaking

As in the case of the input handshaking, the output handshaking is designed to insure that there can be no lost data due to missed signals. PORT 2 is again used for BIU output and the CA2 signal line is used to drive the Graphic 7 OCTL signal line whenever there is valid data on OD00-15. The return handshake signal is processed through the CA1 input to the output VIA.

9.3.2.1 Output Data Valid. When the BIU has a data word to be transferred to the device, a store operation to PORT2A results in the transition of the CA2 signal according to the control specified in the PORT 3 PCR. In the case of the Graphic 7 BIU, a level handshake is used to mimic the standard Harris UBC interface. The signal is held in the high state until the active transition of the return handshake acknowledgment line attached to the CA1 input is detected.

9.3.2.2 Handshake Acknowledgment. As in the case of the Harris BIU, the CA1 signal resets the CA2 output and indicates that the Graphic 7 has accepted the data on the output bus. The OWR signal line from the Graphic 7 is connected to the CA1 input of PORT 2 for this purpose. An active transition on this signal line (low-to-high) also results in the setting of the OWR bit in the P2IFR. This bit is tested in OUTLOP to determine when it is safe to resume transmission of data words.

9.3.3 Deadlock Control

As in the case of the Harris BIU, there is a possibility that the interface could be trying to transmit data in both directions simultaneously. This is the result of the BIU's entering a tight loop to monitor the state of the CA1 signal line. The BIU does not use interrupts from the Graphic 7 to indicate when data is available on the PORT1 bus, therefore, it is possible for the Graphic 7 to be trying to transmit data to the BIU at the same time the BIU is transmitting data to the Graphic 7. This situation is resolved in the same manner as described in Paragraph 8.3.3 but with a shorter delay period of only 600 usecs because the processor has less potential for being busy and thus causing an error.

9.3.4 Sign-Off Processing

The OUTDEV routine examines the message type of each packet it processes. If it is not a data packet, control is transferred to the TRYSOF section of the routine. Here the packet type is compared to the SOFMSG constant to determine if it is a sign-off packet. If it is not, control passes to the TRYSAC section. If it is a sign-off packet the routine checks several flags and variables to detrmine how to process the packet. First the CONECT flag is checked to determine if this BIU is connected to any other BIU (CONECT = +1). If the BIU is connected to another BIU, the address of the BIU sending the sign-off message must be checked to insure that this sign-off message is valid. The third byte of the sign-off packet is compared to the value in XADDR to make this determination. If it is found to be a valid sign-off packet, the value in XADDR is compared to the range of valid addresses for all possible Harris BIUs. A sign-off message from a Harris BIU is probably the result of a power failure and should lead to a sign-on attempt by the previously connected Graphic 7 BIU. If XADDR is within this range of addresses, HARGO is reset to a value which forces execution of the power-up cycle code and control is passed to the RESET routine to reestablish the virtual circuit to the Harris computer. If it was not previously connected to a Harris BIU, the BIU generates a "LINK ENDED." message to the device, followed by the "WHICH SYSTEM?" interrogation. When the code has finished processing the packet, control is passed to the packet freeing section where a call to the DQ subroutine releases it.

9.3.5 Sign-On ACK Processing

In the TRYSAC section, the message type is compared to the constant SACMSG to determine if the packet is an acknowledgment of a sign-on request. If it is not, the code transfers control to the TRYSON section. If it is a sign-on ACK, the state of the WAIT flag is examined to determine if this BIU is waiting for a response to a sign-on request message. If the flag is zero, there may be supplementary data in the packet so the code branches to the output section. If, however, the flag is set, it is cleared and the CONECT flag set to +1 to signify that a valid connection exists. The sign-on counter used by the Harris sign-on loop (See Paragraph 9.4.2) is cleared and a check for output data is made. If the acknowledgment has a length field greater than 7, supplementary data for the device is present. In this case, control is passed to the output section of the routine to send this data. Otherwise, the "LINK OKAY." message is printed on the Graphic 7 display. Here again a determination as to the type of BIU (Serial or Parallel) is made and the TERMF set if found to be a serial device. At this point, the routine is finished processing the data, so control passes to the freeing section of the routine to DQ the packet.

9.3.6 Sign-On Request Processing

The final check of the packet type occurs in the TRYSON section of the routine. If it is not a sign-on request from another BIU, the packet is discarded by the freeing section of the code. Sign-on request packets are valid only if the BIU is not already connected to another BIU. For this reason, the first check in this section of the routine is the state of the CONECT flag. If the BIU is already connected or waiting for the response to a sign-on request, the packet is ignored. If, however, the CONECT and WAIT flags indicate that neither state exists, the "LINK REQUEST FROM" message is transmitted to the device along with the address of the BIU attempting the sign-on. The return address is stored in XADDR and the CONECT flag set to zero to indicate the BIU is waiting for a response to the "LINK REQUEST ... " message. The TYACK timer is set by a call to the STIMER subroutine to prevent a locked situation that can result if the originator's sign-on ACK timer expires before the Graphic 7 user responds to the message. This timer allows the user 10 seconds to respond and then clears the flags (See Paragraph 9.4.2).

9.4 CKTOUT Subroutine

As in the case of the Harris BIU, the Graphic 7 CKTOUT subroutine is used to check several system timers for time-outs. In the Graphic 7 BIU, however, the routines vary greatly from those used in the Harris BIU. In addition to the standard status message timer check, three other checks are made. These checks are for

(1) Sign-On ACK timeout,

(2) Yes-No response timeout, and

(3) TTY Timer timeout.

All three checks are detailed in the paragraphs below and the flow chart for the Graphic 7 CKTOUT appears in Figure 9.4-1.

9.4.1 Sign-On Timeout

During the process of attempting to connect to another BIU, the TSACK timer is set based on the type of device addressed. If a call to CTIME reveals the TOD clock has exceeded the value in TSACK, the code assumes the addressed BIU is not available for the connection. If the sign-on request was to a Harris BIU, the code attempts a sign-on to the next Harris BIU in sequence by incrementing XADDR and generating another sign-on request message in PCONST. The code uses a section of the INDEV subroutine labeled SYSTMX to accomplish this sign-on. The range of valid Harris addresses is searched circularly, beginning with a starting address based on this BIU's home address. The counter SONCNT is used to keep track of the number of sign-on attempts to force an end to the sign-on loop when the total number of Harris BIU, TRMCNT, is exceeded. RSXADR is used to reset the base Harris address if the maximum Harris address is exceeded before SONCNT exceeds TRMCNT. If no system is available for the sign-on, control passes to NOSYS where the "SYSTEM UNAVAILABLE." message is transmitted to the Graphic 7 and control passed to the ASKAGN section of the INDEV routine (See Paragraph 9.2.3.2).

9.4.2 Sign-On Request Timeout

When another BIU has requested a connection to this BIU, the message transmitted to the Graphic 7 must be answered

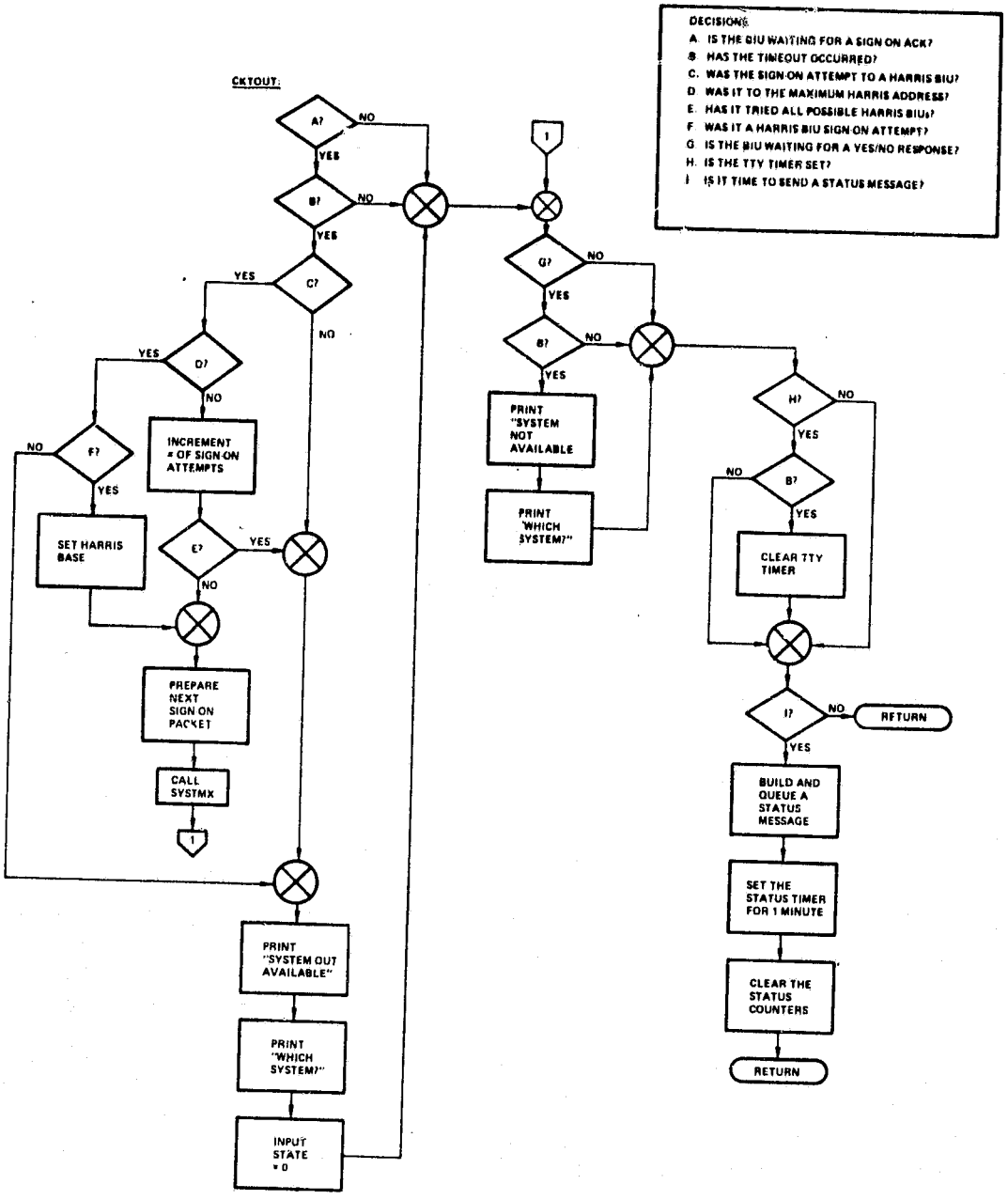


Figure 9.4-1: Graphic 7 BIU CKTOUT Subroutine Flow Chart

within 10 seconds. If it is answered in the affirmative after 10 seconds has passed, the requestor has already stopped waiting for a response but the responder now thinks that there is a valid link. To prevent the BIU from becoming locked requiring the user to press the RESET button, the TYACK timer is set for 10 seconds and the CKTOUT routine used to clear the sign-on ACK flags when necessary. If the call to CTIME reveals that the timer has expired, the "SYSTEM UNAVAILABLE." message is transmitted to the Graphic 7 along with the "WHICH SYSTEM?" interrogation and the CONECT flag set to -1 to free the BIU.

9.4.3 TTY Timout

As was mentioned in the INDEV subroutine (Paragraph 9.2.1), the TTY timer is used to prevent one-word DMA operations from being frozen in the BIU. If a call to CTIME indicates the TETTY variable exceeds the TOD clock, the TTYTM flag is cleared. This forces the INDEV routine to send the buffered packet during the next input data cycle (See Paragraph 7.4.3).

9.5 PUTSTR Subroutine

The PUTSTR routine is used to send service messages generated by the BIU to the Graphic 7 terminal. The main feature of the routine that is unique to the Graphic 7 BIU software is the use of the SNDOUT section of the OUTDEV routine as a subroutine. This is implemented primarily to conserve EPROM storage. The value in the "X" register is held constant to simulate a TTY transaction with the Graphic 7. Each byte of the message is routinely loaded into the "A" register before the call to SNDOUT. When the value loaded from the ASCII table into the "A" register is equal to zero, the end of the message has been reached and control is returned to the calling routine.

9.6 Constants and Tables

The ASCII strings found below PUTSTR in the PROM memory are used by various routines to compare data input or generate message outputs to the Graphic 7. As was mentioned in the discussion of the INDEV routine, the variables in TABLE are used to check for valid sign-on requests from the user. In the Graphic 7 application, currently four possible re-

sponses to the system interrogation message are recognized. The strings following ASCII are used by the various routines transmitting these stored messages to the user. Note that each message is terminated by a null byte equal to zero. Each entry into the table is marked by a label which is based on the difference of its address from ASCII. This allows the indexed operation of the PUTSTR using the "Y" register to point to each byte of the message in turn. As in the Harris BIU, the final 12 bytes of the constants section contain the special status message recognition code. In the case of all Graphic 7 BIUs, this code is "GRAPHIC7 BIU."

PART IV

DIAGNOSTIC PROCEDURES

SECTION X

CABLE SYSTEM TROUBLESHOOTING

10.0 INTRODUCTION

An important function of this document is to provide system support personnel with the ability to diagnose and repair system failures in a timely manner. This section of the report provides these guidelines. The procedures documented here should allow fault isolation and replacement usually in much less than an hour. It is assumed that failures occur rarely enough that reinstatment of the system can be accomplished by replacement of the faulty part. Failures resulting from software errors or contamination of the software by unintended modifications are not addressed.

Failures that can cause interruptions in the CAPS communications can occur at any one of 7 possible sites. These sites are listed in Figure 10.0-1, together with identifying numbers which are used in the diagnostic procedures in Paragraph 10.1.

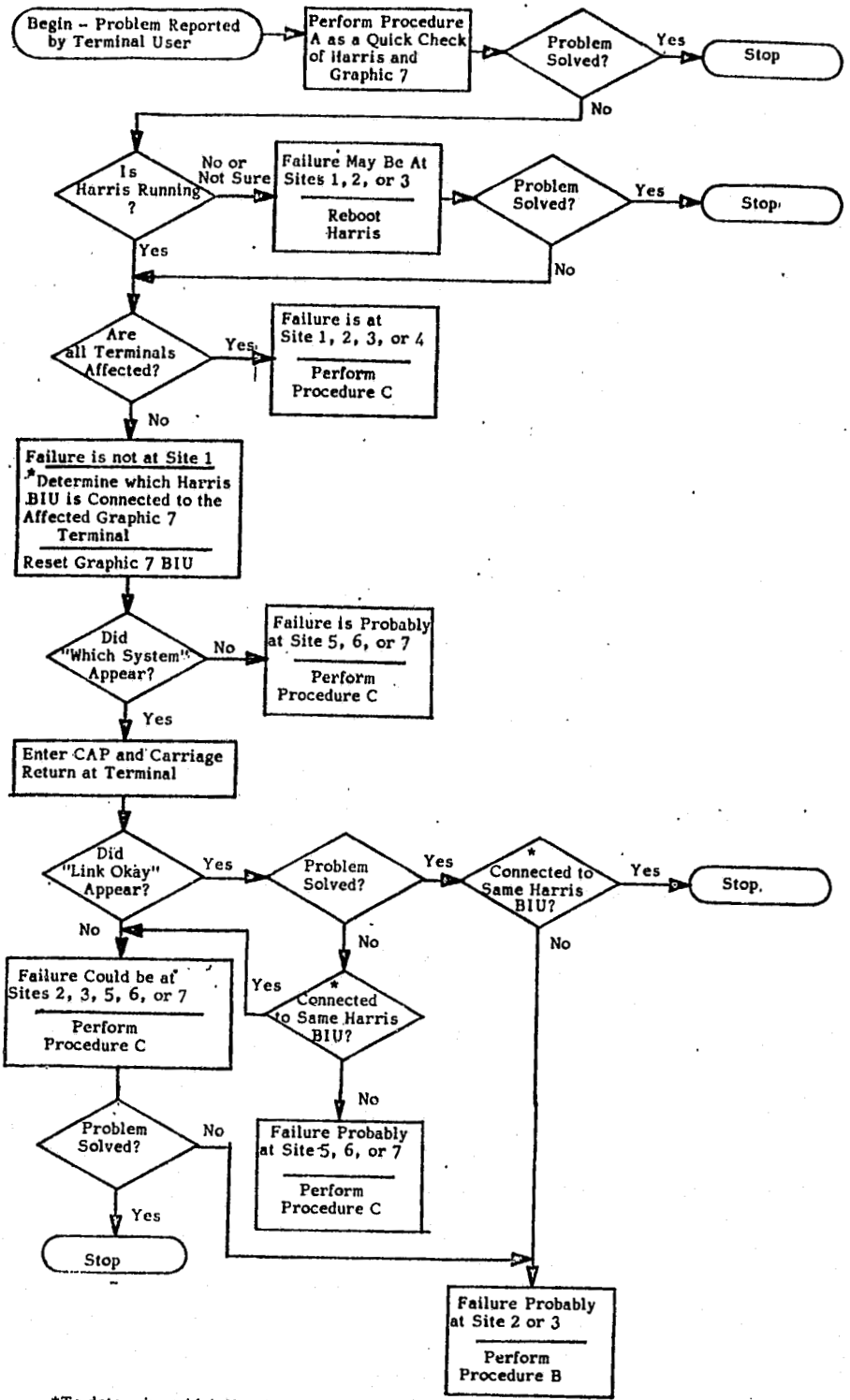
Site Number	Site
1	Harris Handler Software
2	Harris UBC Interface Hardware
3	Harris UBC BIU
4	Radio Frequency (RF) Cable Plant
5	Graphic 7 BIU
6	Graphic 7 Parallel Interface Board
7	Graphic 7 Graphic Control Package (GCP) Software

Figure 10.0-1 Communications Failure Sites in the CAPS

Several diagnostic approaches and tools are available to localize faults in the CAPS communication system. The first is the basic procedure of substituting known good parts (spare) for suspected faulty elements. Special purpose tools available to aid in this isolation and discussed in the following paragraphs include the backboard BIU, a Harris program called VOLLEY and a dummy cable system (test jig).

10.1 Fault Diagnosis Procedures

The diagnostic procedures shown in Figures 10.1-1 through 10.1-7 can be used to systematically isolate elements of the CAPS communication cable system. Figure 10.1-8 is a description of the test jig used in certain tests. Diagnostic testing should begin by following the steps in Procedure A (Figure 10.1-1) and continue through the steps as directed. Component substitution is used where possible, however, detailed instructions on testing or repairing components is beyond the scope of this document. Section XI contains a limited discussion of a suggested approach to the repair of BIUs.



*To determine which Harris BIU is connected to a malfunctioning Graphic 7 BIU, examine the status message generated by the Graphic 7 BIU in the connected address field.

Figure 10.1-1: Top Level Diagnosis Procedure

1. Quick check of the affected terminal:

   a. Verify that power is on the Graphic 7, to the display (note switch light), to the hard copy unit, and to the BIU.

   b. Press Graphic 7 RESET button (note test pattern).

   c. Place Graphic 7 in TTY mode by entering a carriage return followed by a "Y" followed by another carriage return.

   d. Enter CNTL "S" (note BIU XMIT and RCVR Lights should blink when this entry is made). If the lights do not blink, swap the Graphic 7 BIU using Procedure D.

2. Quick check of Harris operation:

   a. Verify that all essential elements have power. If trouble is found, correct problem and continue with normal operation. At least the following elements should be checked:

      (1) Harris BIU
      (2) Harris CPU cabinet
      (3) Cable system power supply
      (4) Harris system control console
      (5) Harris disks

   b. Check that the Harris is not hung and that the control console responds to inputs. If the Harris is hung, reboot it.

Figure 10.1-2: Diagnostic Procedure A - Quick Check of User Terminal and of Harris

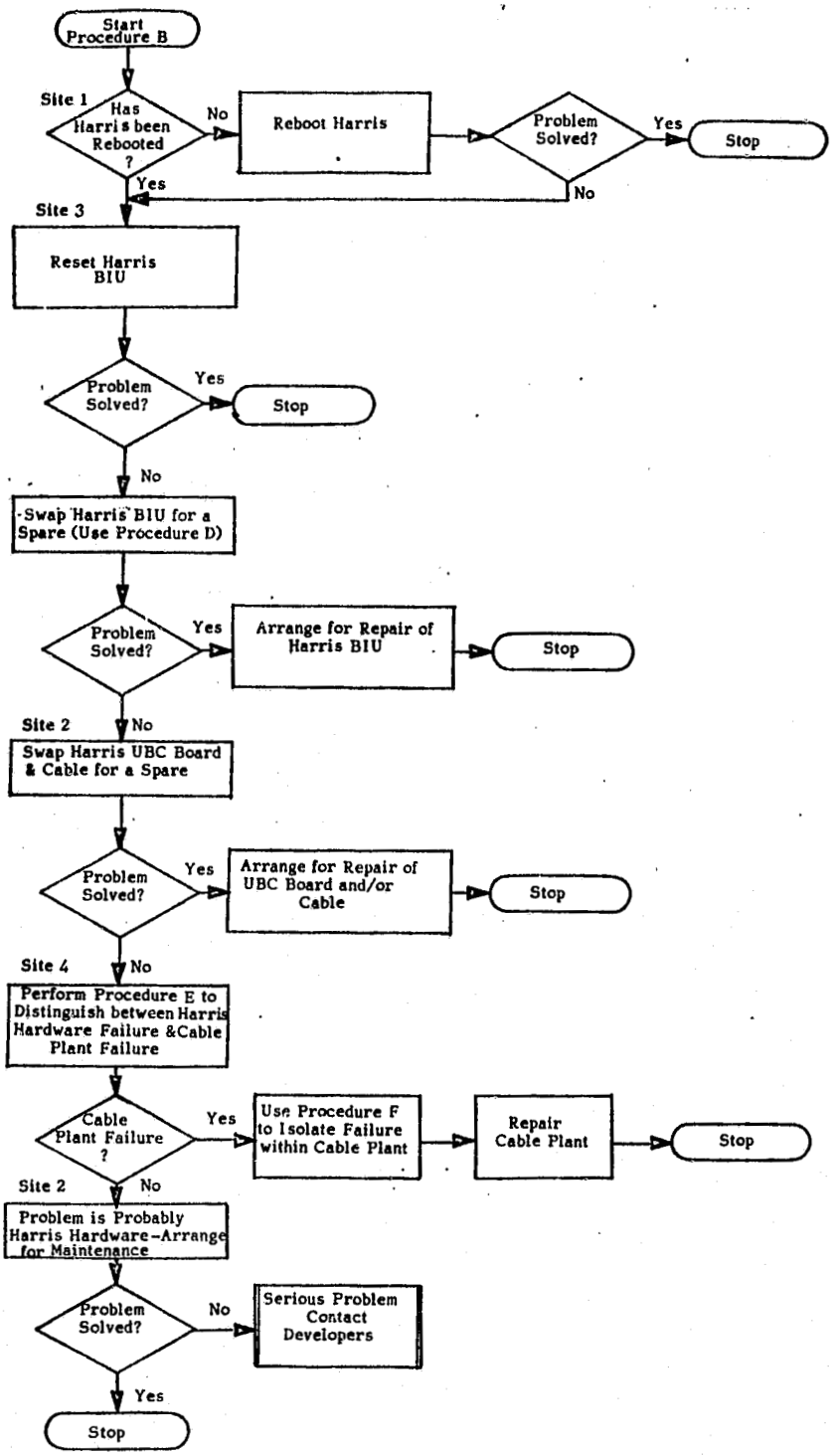


Figure 10.1-3: Diagnostic Procedure B - Eliminates Sites 1, 2, 3 and 4 (Harris Software and Hardware, Harris BIU and Cable Plant)

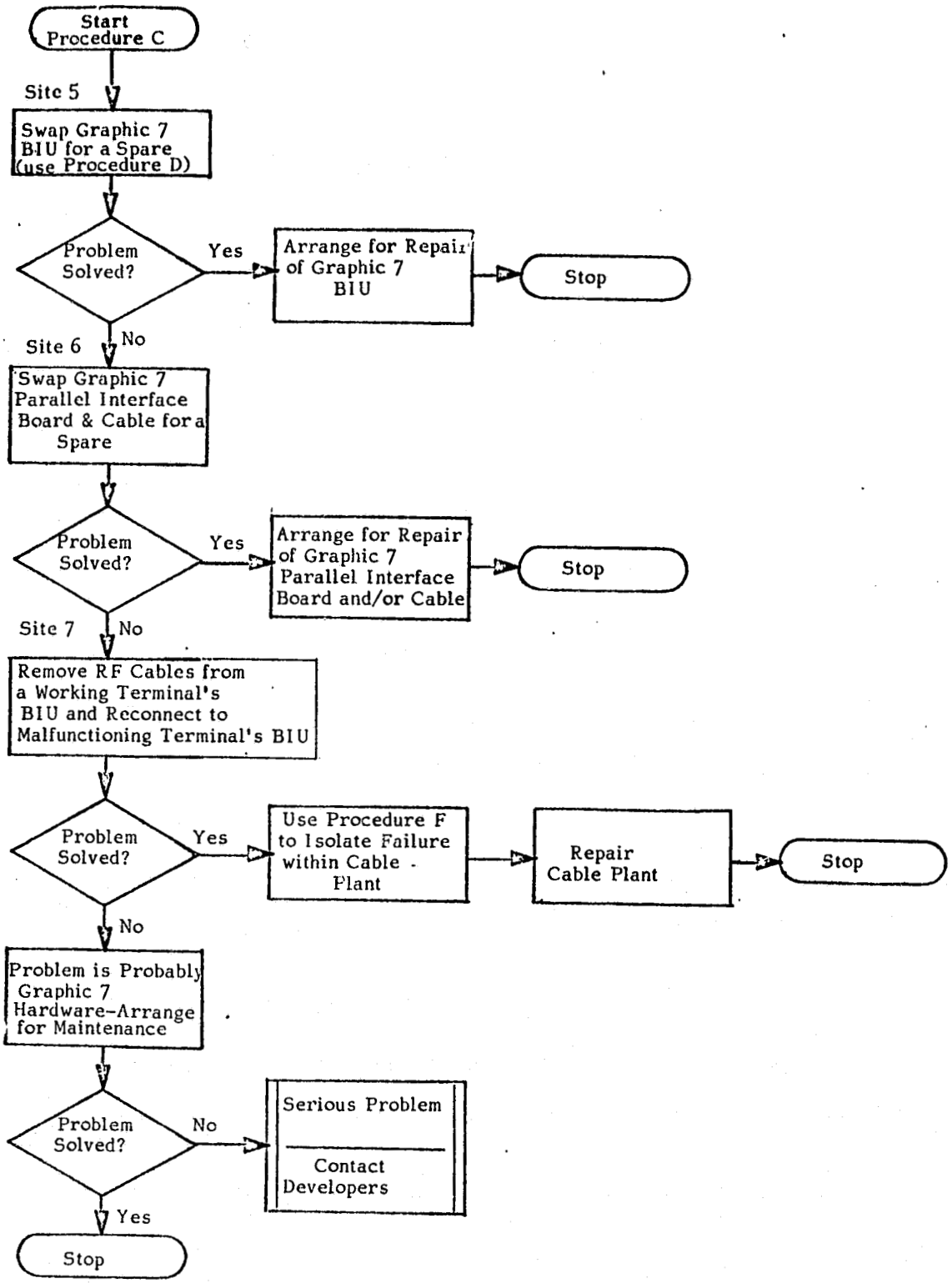


Figure 10.1-4: Diagnostic Procedure C - Eliminates Sites 5, 6 and 7 (Graphic 7 BIU, Graphic 7 Software and Hardware)

1. Turn off power to the BIU.

2. Unplug power cord from the BIU and disconnect the parallel cable at the BIU.

3. Move Programmable Read-Only Memory (PROM) chips from old BIU to new BIU, if necessary. NOTE: When substituting similar operational BIUs from another location, this step is normally not necessary.

   a. Remove top cover of BIU by removing four attaching screws on the bottom of the unit.

   b. The PROMs are located at board coordinates U22 and R14 (See Table VI). Carefully lift up on the chips using a chip extraction tool insuring that the pins are not bent.

   c. Carefully insert the chips in the replacement BIU insuring that the PROM labeled F800 in at coordinate location U22 and the PROM labeled FC00 is at location R14. NOTE: When the chips are properly inserted, there will be one empty pin socket at each end of each PROM.

4. Move the RF cables from the old BIU to the replacement BIU. The orange or red banded cable should be connected to the transmit modem (marked with a "T" on the back of the BIU), while the unbanded cable should be connected to the receive modem ("R").

5. Connect the parallel cable to the new BIU and connect the power cord to the new BIU.

6. Turn on power to the new BIU.

Figure 10.1-5: Diagnostic Procedure D - Method of Swapping BIUs

NOTE: One Graphic 7 terminal cannot be used while this test is in progress because it uses one Harris BIU and the UBC associated with it.

1. Build or locate the bus test jig, which is constructed as follows:

   Two four-way multitaps (with 20 dB attenuation to tap -- Jerrold FFT-4-20 or equivalent) are connected back-to-back using a connector such as the Jerrold VHH. The pass-through outputs on each tap (See Figure 10.1-8) should be terminated with 75-ohm resistive terminators (such as the combination of Jerrold VSF-59A adapter and TR-75F terminator). The arrows embossed into the tap cases should point away from each other. Designate one of the taps as the transmit tap.

2. Connect the Harris BIU to be tested and the backboard diagnostic BIU to the test jig by attaching the transmit modem of each BIU to the transmit tap and the receive modem of each BIU to the other tap. NOTE: Any CAPS BIU can function as the backboard by inserting the backboard PROMs using Procedure D.

3. At any Harris TTY terminal sign-on using any valid access code and enter the command "VOLLEY XX" to execute the program where XX is the physical device number (PDN) of the Harris UBC to be used for the test.

4. VOLLEY will then begin writing messages at the terminal and in the List Output (LO) file. If the message "BACKBOARD IS OKAY; hh:mm:ss.ttt mm/dd/yyyy appears, the CAPS problem is in the cable plant (site 4). If other messages appear, the problem is in the Harris hardware (site 2) which affects writing to the BUS.

5. Remove the VOLLEY program by executing a program abort (Shift L at a dumb terminal or CNTL S at a Graphic 7).

Figure 10.1-6: Diagnostic Procedure E - Distinguishes between Sites 2 and 4 (Harris Hardware and Cable Plant)

NOTE: All steps of this procedure can be performed concurrently with regular CAPS work if a portion of the cable system is still operational.

1. Perform a quick check of the bus system power supply at the headend (located in a 19" cabinet in Room 274 Building 4).

   a. Check that the power supply is receiving 115 VAC from the outlet.

   b. Check that the power indicator light is on.

   c. Check the power supply circuit breaker inside the chassis and the power combiner fuses by removing the cover of the power combiner (See Figure 3.1-2).

2. Install the cable test PROMs in a serial BIU which is connected to any serial I/O device (i.e. TI Silent 700, Graphic 7 without parallel interface card, etc.) in accordance with Procedure D setting the BIU's data rate to that of the device using Table XII. Connect the BIU to the cable at the headend. Bus failures can be isolated by following the steps given below. The locations referenced in the steps are keyed to the schematic diagram of the cable system shown in Figure 10.1-9 (More information about the cable system can be found in [12]).

   In each of the steps below, first connect the backboard BIU (any BIU with the backboard diagnostic PROMs installed) to the cable system at the indicated test point; the transmit cable from the backboard should go to the odd-numbered point (orange-banded cable) and the receive cable to the even-numbered point (green-banded cable). The test is initiated by resetting the cable test BIU and entering the date and time. If the message "BCKBRD OK. mm/dd/yy hh:mm:ss TEST BYTE IS X" is printed

Figure 10.1-7: Diagnostic Procedure F - Isolates and Eliminates Failures within Site 4 (Cable Plant)

on the terminal once each minute, the cable plant is operational between the headend and the test point being checked.

NOTE: It is possible to reposition the backboard without resetting the cable test BIU. While the backboard is not connected to the bus the message "NO BCKBRD ECHO. mm/dd/yy hh:mm:ss" will appear at the terminal.

For ease in checking, conduct the tests in the following manner:

(1) Test points 1 and 2 (Only taps in Building 4)

(2) If points 1 and 2 are operational, test the remaining points in order.

(3) Use Table XI to determine in which section of the cable plant the failure may be located.

3. When the failed portion of the system has been isolated, bus communications should be restored by the following means.

[The portion of the cable containing the failure may include multitaps, line amplifiers, and/or directional couplers, as well as cable and connectors. Repairs of these components are discussed in the following steps.]

NOTE: POWER TO THE CABLE SYSTEM MUST BE OFF BEFORE ANY OF THE FOLLOWING STEPS ARE TAKEN.

a. Check the connections to the cable at suspected problem points. Make certain that the center conductor is firmly grasped by the set screw in all box-type components but not so tightly that the center conductor is severed. Check that

Figure 10.1-7: (Continued)

the braid or sheild of the cable is firmly grasped by the connector.

b. If the failed component is a multitap, replace it with a spare of the same value; tap values are shown in Figure 10.1-9. Note that the shell of the tap can remain in place and only the internal portion substituted. If a tap has failed such that it affects only terminals connected to it, the tap may be left inplace and the terminals connected to another convenient set of taps. The transmit and receive cables of a terminal should normally be connected to taps in the same pair (as shown in Figure 10.1-9).

c. If the failed component is a line extender amplifier, replace the internal amplifier unit (leave the shell in place) and realign the amplifier using the procedure given in [12]. Make certain that the amplifier power tap is on the 45-60V range and the THRU/STOP switch is in the same position as the unit being replaced.

d. If the failed component is a directional coupler, replace the interior part of the directional coupler (leave the shell in place) with a coupler of the same isolation (as shown in Figure 10.1-9).

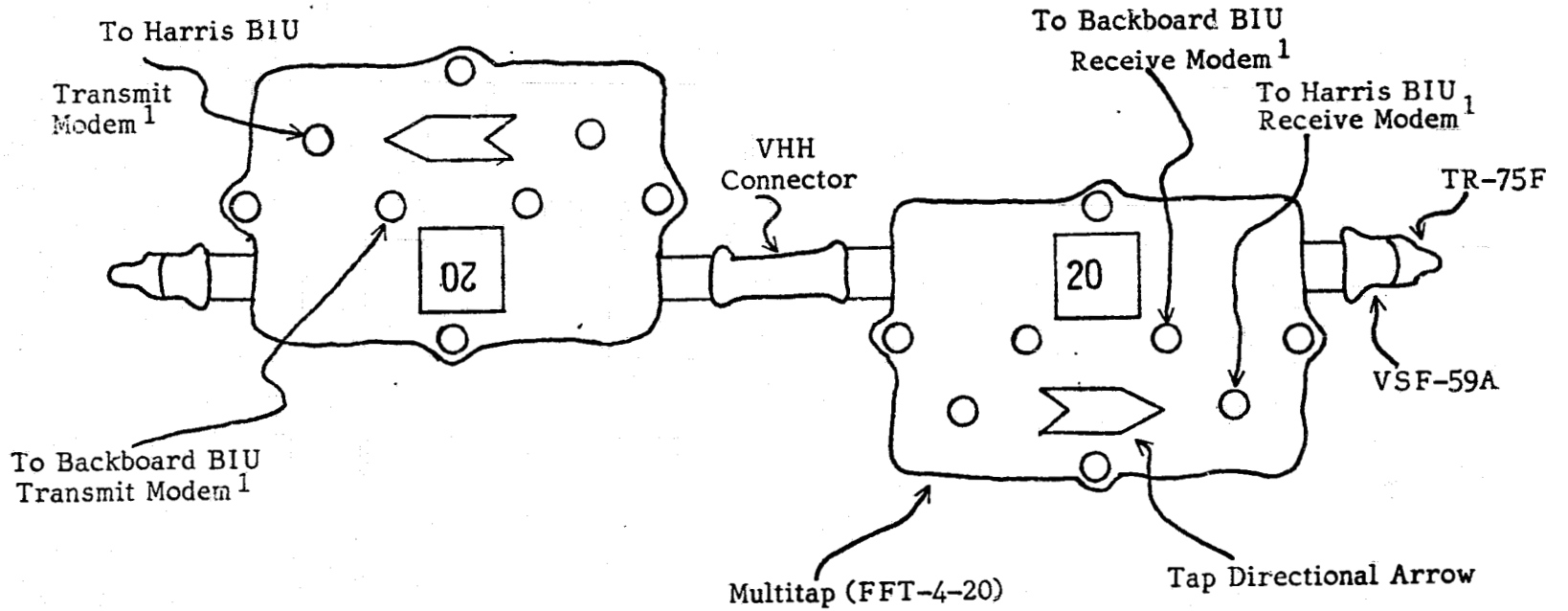
e. In the unlikely event that the failure is found to be an actual open or short circuit in the cable itself, either the entire cable piece which has failed can be replaced, or the damaged area can be cut out and a new piece added using universal box splices (such as Jerrold PBA Series). Location of the exact location of a break is beyond the scope of this report. When installing any new cable, insure that connections are made (step a. above) with the braid and the center conductor making a good continuous electrical circuit.

Figure 10.1-7: (Concluded)

TABLE XI

CABLE PLANT FAILURE ISOLATION

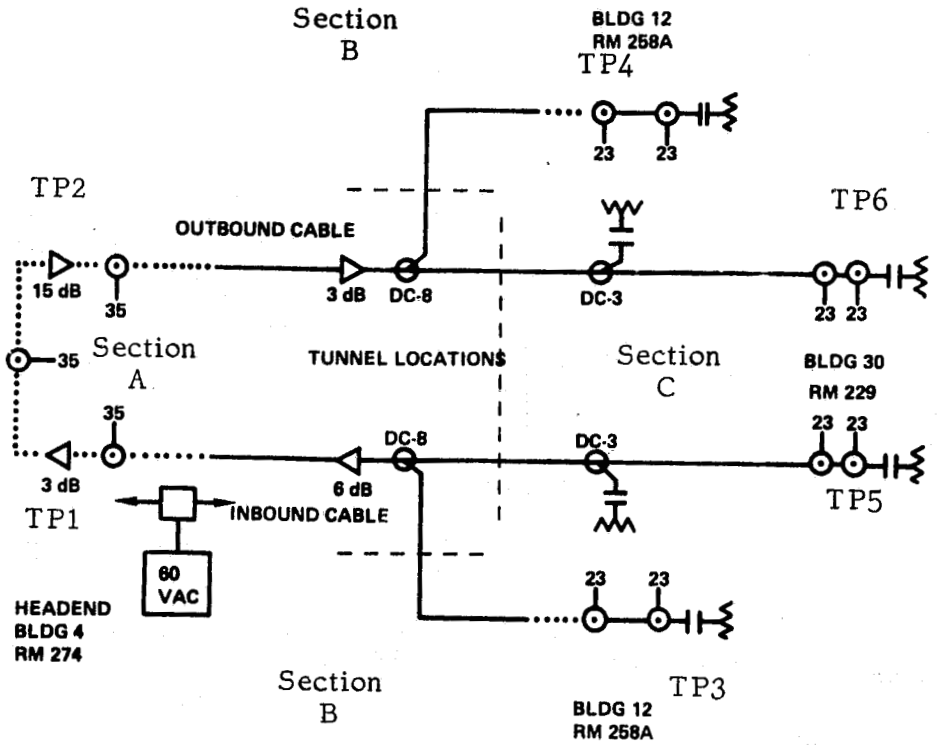
Points of Failure	Cable Sections Affected
3 & 4 Only	Section B Only
5 & 6 Only	Section C Only
3,4,5 & 6	Section A and/or Both Sections B and C
ALL	Go back to Step 1, Procedure F



Note 1: Any of the 4 RG-59 Connections on this tap may be used.

Figure 10.1-8 Structure of Bus Test Jig

IB-56,736



NOTES:
1. VALUES ADJACENT TO DIRECTIONAL COUPLERS DENOTE ISOLATION FROM INPUT TO TAP.
2. VALUES ADJACENT TO AMPLIFIERS ARE ATTENUATION PAD VALUES.
3. VALUES ADJACENT TO MULTITAPS DENOTE ISOLATION FROM INPUT TO TAP.
4. ALL MULTITAPS HAVE 4 OUTPUTS.

Figure 10.1-9: CAPS Cable Network Schematic

SECTION XI

BUS INTERFACE UNIT TROUBLESHOOTING

11.0 INTRODUCTION

Repair of Bus Interface Units should ordinarily be undertaken by qualified engineers or technicians who work from the schematic diagrams and explanations in Section V to troubleshoot the digital logic of the BIU. The BIU naturally falls into a modular division, however, and in many cases simple replacement of modules provides an easy way to isolate a problem. Operating experience has shown that the most likely components to fail in a BIU are logic chips on the digital board, the modular power supply, and circuitry (usually a chip) in either the modulator or the demodulator. The remainder of this section defines several steps that may be useful in dealing with BIU problems in these areas.

11.1 Diagnosis of BIU Trouble

As discussed in Section V, the digital board of every CAPS BIU is capable of supporting a parallel interface or a serial interface, but the BIU chassis contains only the parallel CAPS connector. Standard serial BIUs are available from the TMS system and can serve to help isolate problems on the digital board. The BIU's function is defined by the PROM chips which are inserted into the digital board. Operationally, the circuitry of the BIU falls into sections associated with support of parallel interfaces, serial interfaces, network interface and the CPU operation. Some types of failures affect all four operational portions of the BIU, while others affect only one.

The testing of the network and serial sections of the BIu can be accomplished on an operational bus system without interfering with normal users, provided that an additional BIU (which must be a serial BIU with PROMs to support an asynchronous terminal) and a "dumb" terminal with RS-232C interface are available. The second BIU should be connected to the bus system (the transmit modem ["T"] should be connected to the orange-banded cable and the receive modem to the unbanded cable), and the "dumb" terminal should be connected to the BIU's RS-232C plug. The red dual-inline-package (DIP) switch on the BIU's digital board at location

N24 (See Table VI) should be set to agree with the speed of the asynchronous terminal. The switch package is marked on one side with the switch numbers and on the other side with the word "OPEN." Exactly one switch position should be depressed on the side with the numbers; all other positions should be depressed on the side marked "OPEN." Table XII shows the corresponding terminal speeds indicated by the switch positions.

TABLE XII

DIP SWITCH SETTINGS FOR ASYNCHRONOUS SERIAL TERMINALS

Switch Position	Speed (Baud)
8	75
7	150
6	300
5	600
4	1200
3	2400
2	4800
1	9600

The test arrangement of the serial BIU and "dumb" terminal can be left connected to the network indefinitely (either powered or not), if desired.

Figure 11.1-1 shows a diagnostic procedure that can be used to isolate failed portions of a BIU. It should be noted that this procedure will not identify all possible problems, but has been designed to pinpoint quickly the most probable failures.

1. Perform a quick check for power in the BIU by first checking the fuse (next to the power cord connection) and power supply output voltages.

   If the power supply connector pins are numbered from right to left when viewed from the top front of the BIU, then the following voltages should be present:

Pins	1,2	115 VAC
	3	-5 VDC
	4	-12 VDC
	5	+12 VDC
	6-8	GND
	9,10	+5 VDC

2. To isolate the failed section of the malfunctioning BIU, perform the following tests:

   a. Install the backboard PROMs in the bad BIU and connect the BIU to the bus system. Connection of the BIU and insertion of the PROMs should be done as described in Diagnostic Procedure D (Figure 10.1-5).

   b. Press the RESET button on the "dumb" terminal BIU, and answer the "WHICH SYSTEM?" query with "BAC" (followed by a carriage return).

   c. If "LINK ACCEPTED appears at the "dumb" terminal, type a string of 20 or so arbitrary characters (followed by a carriage return) and verify that the same string is echoed back from the backboard. If "LINK ACCEPTED" does not appear or this test fails, go to step e.

Figure 11.1-1: Diagnostic Procedure for Failed BIUs

d. If this test passed, then the most probable failure is a chip on the BIU digital board; most probably a 6522 chip. Chip substitution of the 6522 chips with known good chips should be attempted at this point. The BIU can then be tested in the CAPS system once again. If the BIU malfunctions, repair attempts should be continued first by replacing the parallel circuit drivers, the 74S240 chips, and then by digital logic checks using the information in Section V.

e. If the bad BIU fails the backboard test in step c. above, remove the power to the BIU and substitute a known good digital board (move the backboard PROMs to the new board). Now retry the test in step c. If the test succeeds, the problem is probably a bad chip on the first digital board; chips should be replaced systematically before detailed logic checkout is begun (using information found in Section V).

f. If substituting digital boards, as in step e., does not correct the problem, substitute known good modems in the bad BIU. In performing this substitution, substitute modems in pairs only and make certain that the transmit modem (marked with a "T") is inserted into the slot marked with a "T" and the receive modem (marked with an "R") is inserted into the slot marked with an "R". REVERSING THE MODEM POSITIONS WILL DAMAGE THE MODEMS.

   After substituting the modems, retry the test in step c. If the test succeeds, the problem lies in one of the modem boxes. Documentation in [13] will be useful in determining the exact failure.

g. If the test of step f. fails, the problem most likely lies in the BIU power supply or possibly in a connection failure in the chassis.

Figure 11.1-1: (Concluded)

APPENDIX I

HARRIS BIU SOURCE LISTING

This is the source listing for the Harris BIU software. The reader is directed to Sections VII and VIII for a detailed description of the operation of this code. The code used in this software is the standard assembly language for the MCS 6500 family of microprocessors found in [23].

```
CARD #  LOC   CODE      CARD  10        20        30        40        50        60        70
     2  0000              .OPT NOCNT,XREF,MEM,LIST,ERR,GEN
     3  0000             ;
     4  0000             ; THIS IS THE HARRIS BIU CODE AS OF 12/11/79 WITH FLOW CONTROL
     5  0000             ; ****** NOTE: CODE HAS BEEN ADDED TO ALLOW OPERATION OF THE VOLLEY  ******
     6  0000             ; ******       TEST PROGRAM AND THE ACKNOWLEDGMENT OF STATUS MESSAGES. ******
     7  0000             ; ******       IN ADDITION, THIS PROGRAM ALLOWS USE OF SWITCH          ******
     8  0000             ; ******           SELECTABLE MAIN ADDRESS                             ******
     9  0000             ;
    10  0000             STATUS   = $00          STATUS MESSAGE BUS ADDRESS
    11  0000             HARADW   = $0F          HARRIS ADDR WITH WRITE FLAG
    12  0000             HARADR   = $2F          HARRIS ADDR WITH READ FLAG USED DURING CNCT CYCLE
    13  0000             TRMBAS   = $51          THIS IS THE BASE ADDRESS FOR G7 TERMINALS
    14  0000             TRMCNT   = $0F          THE NUMBER OF G7 TERMINALS IN THE SYSTEM
    15  0000             ; MESSAGE TYPE EQUATES
    16  0000             ;
    17  0000             DATMSG   = 0            DATA MESSAGES ARE 00-1F
    18  0000             STAMSG   = $DB          STATUS MESSAGE
    19  0000             SOFMSG   = $DE          SIGN-OFF MESSAGE
    20  0000             SACMSG   = $DF          SIGN-ON ACKNOWLEDGMENT
    21  0000             SONMSG   = $E0          SIGN-ON MESSAGE (E0-FF)
    22  0000             TTY      = 2            DATA TYPE FOR TTY-COMPATIBLE TERMINALS
    23  0000             ;
    24  0000             ; DEVICE ADDRESSES FOLLOW
    25  0000             ;
    26  0000             NUARTS   = $C00         NETWORK UART STATUS
    27  0000             NUARTD   = $C01         NETWORK UART DATA
    28  0000             DUARTS   = $1400        DEVICE UART STATUS
    29  0000             PORT1B   = $1010        PARALLEL PORT 1B DATA REGISTER
    30  0000             PORT1A   = $1011        PARALLEL PORT 1A DATA REGISTER
    31  0000             P1B3DR   = $1012        PARALLEL PORT 1B DATA DIRECTION REGISTER
    32  0000             P1A3DR   = $1013        PARALLEL PORT 1A DATA DIRECTION REGISTER
    33  0000             T1#RL    = $1014        TIMER LOW BYTE
    34  0000             T1#RH    = $1015        TIMER HIGH BYTE
    35  0000             P1ACR    = $1018        PARALLEL PORT 1 AUXILIARY CONTROL REGISTER
    36  0000             P1PCR    = $101C        PARALLEL PORT 1 PERIPHERAL CONTROL REGISTER
    37  0000             P1IFR    = $1010        PARALLEL PORT 1 INTERRUPT FLAG REGISTER
    38  0000             P1IER    = $101E        PARALLEL PORT 1 INTERRUPT ENABLE REGISTER
    39  0000             PORT2B   = $1020        PARALLEL PORT 2B DATA REGISTER
    40  0000             PORT2A   = $1021        PARALLEL PORT 2A DATA REGISTER
    41  0000             P2B3DR   = $1022        PARALLEL PORT 2B DATA DIRECTION REGISTER
    42  0000             P2A3DR   = $1023        PARALLEL PORT 2A DATA DIRECTION REGISTER
    43  0000             P2ACR    = $1028        PARALLEL PORT 2 AUXILIARY CONTROL REGISTER
    44  0000             P2PCR    = $102C        PARALLEL PORT 2 PERIPHERAL CONTROL REGISTER
    45  0000             P2IFR    = $1020        PARALLEL PORT 2 INTERRUPT FLAG REGISTER
    46  0000             P2IER    = $102E        PARALLEL PORT 2 INTERRUPT ENABLE REGISTER
    47  0000             PORT3B   = $1040        PARALLEL PORT 3B DATA REGISTER
    48  0000             PORT3A   = $1041        PARALLEL PORT 3A DATA REGISTER
    49  0000             P3B3DR   = $1042        PARALLEL PORT 3B DATA DIRECTION REGISTER
    50  0000             P3A3DR   = $1043        PARALLEL PORT 3A DATA DIRECTION REGISTER
    51  0000             P3ACR    = $1048        PARALLEL PORT 3 AUXILIARY CONTROL REGISTER
    52  0000             P3PCR    = $104C        PARALLEL PORT 3 PERIPHERAL CONTROL REGISTER
    53  0000             P3IFR    = $1040        PARALLEL PORT 3 INTERRUPT FLAG REGISTER
    54  0000             P3IER    = $104E        PARALLEL PORT 3 INTERRUPT ENABLE REGISTER
```

```
CARD #  LOC   CODE      CARD  10        20        30        40        50        60        70
   56   0000           ;
   57   0000           ; EQUATES FOR PORT 1 HANDSHAKING FLAGS IN P1IFR
   58   0000           ;
   59   0000           ;       *** ODACP IS THE SIGNAL FROM CA2 OF 6522/1 TO SIGNAL ***
   60   0000           ;       ***  THE BIU HAS TAKEN HARRIS DATA OR COMMAND INFO   ***
   61   0000           ;
   62   0000           ODH     = $02           1 => HARRIS HAS OUTPUT DATA FOR BIU (OCTL)
   63   0000           CDH     = $10           1 => HARRIS HAS COMMAND DATA FOR BIU
   64   0000           ;
   65   0000           ; EQUATES FOR PORT 2 HANDSHAKING FLAGS IN P2IFR
   66   0000           ;
   67   0000           ;       *** DAVFU IS THE SIGNAL FROM CA2 OF 6522/2 TO SIGNAL ***
   68   0000           ;       ***  THE BIU HAS DATA FOR THE HARRIS                  ***
   69   0000           ;
   70   0000           DATU    = $02           1 => HARRIS HAS TAKEN BIU DATA (ICTL)
   71   0000           ;
   72   0000           ; EQUATES FOR PORT 3 INPUT/OUTPUT SIGNAL LINES
   73   0000           ;
   74   0000           ;       *** CNCT IS THE SIGNAL ON CA2 OF 6522/3 TO SIGNAL   ***
   75   0000           ;       ***  THE BIU IS CONNECTED TO THE HARRIS              ***
   76   0000           ;
   77   0000           DISC    = $02           HARRIS SIGNAL TO BIU TO READ UCR
   78   0000           IIFU    = $08           INPUT INTERRUPT OF HARRIS
   79   0000           NIIFU   = $F8           TURN OFF INPUT INTERRUPT
   80   0000           ;
   81   0000           ; THE NEXT EQUATES REFLECT THE ACTUAL OPERATIONAL PARAMETERS
   82   0000           ; OF THE BIU WITH RESPECT TO BUFFER COUNT AND SIZE
   83   0000           ;
   84   0000           RAMSIZ  = 3072          NUMBER OF BYTES OF RAM AVAILABLE
   85   0000           BUFLEN  = 128           NUMBER OF BYTES IN A BUFFER (DO NOT CHANGE)
   86   0000           BUFMEM  = RAMSIZ-256    AMOUNT OF MEMORY ALLOCATED TO BUFFERS
   87   0000           BUFCNT  = BUFMEM/BUFLEN NUMBER OF BUFFERS AVAILABLE
```

```
CARD #  LOC   CODE    CARD  10        20        30        40        50        60        70
    89  0000          ;
    90  0000          ; THE FOLLOWING VARIABLES ARE CLEARED WHENEVER RESET IS HIT.
    91  0000          ;
    92  0000  00 00   INTPTR  .DBYTE 0      BUFFER PTR USED BY IRQ
    93  0002  00       INTBC   .BYTE  0      BYTE COUNT USED BY IRQ
    94  0003  00 00   INPTR   .DBYTE 0      BUFFER PTR USED BY INDEV
    95  0005  00       INBC    .BYTE  0      BYTE COUNT USED BY INDEV
    96  0006  00 00   OUTPTR  .DBYTE 0      BUFFER PTR USED BY OUTDEV AND NET
    97  0008  00       OUTBC   .BYTE  0      BYTE COUNT USED BY OUTDEV AND NET
    98  0009  00       OUTPL   .BYTE  0      LENGTH OF PACKET BEING SENT TO DEVICE
    99  000A  00 00   BIJPTR  .DBYTE 0      BIJ-CREATED BUFFER POINTER
   100  000C  00       CURNET  .BYTE  0      THE CURRENT PACKET BEING READ IN FROM THE NET
   101  000D  00       CURDEV  .BYTE  0      THE CURRENT PACKET COMING FROM THE DEVICE
   102  000E  00       CURBIJ  .BYTE  0      THE CURRENT PACKET COMING FROM THE BIU
   103  000F  00       INSET   .BYTE  0      WHETHER A DEVICE INPUT BUFFER IS SET UP
   104  0010  00       OUTSET  .BYTE  0      WHETHER A DEVICE OUTPUT BUFFER IS SET UP
   105  0011  00       INTSET  .BYTE  0      WHETHER A NETWORK INPUT BUFFER IS SET UP
   106  0012  00       RANDJ   .BYTE  0      CURRENT RANDOM NUMBER
   107  0013  00       RNDCNT  .BYTE  0      CURRENT RANDOM COUNT
   108  0014  00       XMIT    .BYTE  0      TRANSMIT FLAG
   109  0015  00 00   TOKEN   .DBYTE 0      TEMPORARY RECEIVED DATA BUFFER
   110  0017  00       MAXPAK  .BYTE  0      THE CURRENT MAX # OF CHARS IN A BUFFER
   111  0018          TOD     *=*+3         THE CURRENT TIME OF DAY IN SECONDS
   112  001B          TSTAT   *=*+3         TIME TO XMIT THE NEXT STATUS MSG
   113  001E          TSACK   *=*+3         TIMEOUT ON ANSWER TO SIGN-ON REQUEST
   114  0021  00 00   NMXMT   .DBYTE 0      # OF PACKETS TRANSMITTED SUCESSFULLY
   115  0023  00 00   NMRXMT  .DBYTE 0      # OF PACKETS RE-TRANSMITTED BECAUSE NO ACK
   116  0025  00 00   NMCOL   .DBYTE 0      # OF COLLISIONS AND RETRANSMISSIONS
   117  0027  00 00   NMDISC  .DBYTE 0      # OF PACKETS DISCARDED
   118  0029  00 00   NMGCRC  .DBYTE 0      # OF PACKETS RECEIVED WITH GOOD CRC
   119  002A  00 00   NMBCRC  .DBYTE 0      # OF PACKETS RECEIVED WITH BAD CRC
   120  002D  00 00   NMLOST  .DBYTE 0      # OF PACKETS LOST BECAUSE OF NO BUFFER SPACE
   121  002F  00 00   NMWAIT  .DBYTE 0      # OF TIMES BIU HAD TO WAIT TO XMIT
   122  0031  00       PARITY  .BYTE  0      USED TO FLAG BAD PARITY IN AN INCOMING PACKET
   123  0032  00       NTPRX   .BYTE  0      # OF TIMES THE CURRENT PACKET IS RETRANSMITTED
   124  0033  00       SQNUM   .BYTE  0      THE SEQUENCE # OF THE NEXT PACKET TO BE XMITTED
   125  0034  00       LASTTX  .BYTE  0      THE XMIT ADDR OF THE LAST PACKET RECEIVED
   126  0035  00       LASTSQ  .BYTE  0      THE SEQUENCE # OF THE LAST PACKET RECEIVED
   127  0036  00       STKPTR  .BYTE  0      INDEX OF TOP OF FREE BUFFER STACK
   128  0037  00       MSXMIT  .BYTE  0      # OF MILLISECONDS TO STOP XMITTING
   129  0038  00       INTX    .BYTE. 0      VALUE OF REGISTER X WHEN NINT IS NEXT ENTERED
   130  0039  00       INTFLG  .BYTE. 0      BYTE CLEARED BY NINT EVERY TIME IT IS ACTIVE
   131  003A  00       WAIT    .BYTE  0      WAITING FOR SIGN-ON ACK FLAG
   132  003B  00       BYTLOW  .BYTE  0      USED TO SAVE THE LOW ORDER BYTE OF AN INPUT
   133  003C  00       DMATM   .BYTE  0      TIMER FOR DMA OPERATIONS
   134  003D  00       ACKBYT  .BYTE  0      BYTE USED TO INDICATE ADDRESS OF ACK PACKET
```

```
CARD #  LOC   CODE    CARD  10        20        30        40        50        60        70
  136   003E          ;
  137   003E          ; THE FOLLOWING VARIABLES ARE NOT CLEARED AT RESET TIME
  138   003E          ; (ALL VARIABLES PRECEDING "XADDR" WILL BE SET TO ZERO.)
  139   003E          ;
  140   003E  00      XADDR   .BYTE  0        THE CURRENT XMIT ADDRESS
  141   003F  00      HARGO   .BYTE  0        INITIAL POWER-UP FLAG
  142   0040  00      CONECT  .BYTE  0        1 => CONNECTED, 0 => WAITING FOR REPLY TO ...
  143   0041          ;     LINK REQUEST MESSAGE, -1 => WAITING FOR REPLY TO WHICH SYSTEM? MSG
  144   0041  00      TICK    .BYTE  0        COUNTS THE # OF CLOCK TICKS IN 1/4 SECOND (25)
  145   0042          ;
  146   0042          ; THE BUFFERS EACH HAVE AN ENTRY IN THIS ARRAY
  147   0042          ;
  148   0042          NEXT    *=**+BUFCNT     THE NEXT POINTER FOR EACH BUFFER
  149   005B          ;
  150   005B          ; THE QUEUE HEADERS FOLLOW
  151   005B          ;
  152   005B  00      QIN     .BYTE  0        THE QUEUE OF BUFFERS WAITING FOR THE HARRIS
  153   0059  00      QOUT    .BYTE  0        THE QUEUE OF BUFFERS WAITING FOR THE NETWORK
  154   005A  00      DMATTY  .BYTE  0        FLAG TO INDICATE MODE OF INPUT OPERATION
  155   005B          ;
  156   005B          ; THE BUFFER ALLOCATION STACK, BELOW, CONTAINS THE FREE BUFFER LIST
  157   005B          ;
  158   005B          BUFSTK  *=**+BUFCNT     THE FREE BUFFER STACK
  159   0071          ;
  160   0071          ; THESE ARE THE POINTERS TO THE BUFFERS
  161   0071          ;
  162   0071          LOPTR   *=**+BUFCNT     THE LOW HALF OF THE PTRS
  163   0087          HIPTR   *=**+BUFCNT     THE HIGH HALF OF THE PTRS
  164   009D          ;
  165   009D          ; THE BUFFERS ARE LOCATED IN CONTIGUOUS MEMORY STARTING WITH BUFFER NUMBER 2
  166   009D          ; AT ADDRESS 0180.  BUFFER NUMBER 1 IS LOCATED IN A GAP BETWEEN THE ZERO PAGE
  167   009D          ; VARIABLES AND THE PROCESSOR STACK (00C0-013F).
  168   009D          ; BUFFER ADDRESSES ARE LOADED INTO HIPTR/LOPTR BY THE RESET ROUTINE.
  169   009D          ;
  170   009D          ; THE PROCESSOR STACK IS SET TO START AT MEMORY LOCATION 017F AND WILL
  171   009D          ; DECREMENT WHEN NECESSARY.  IT WILL USE RAM AREA BETWEEN 017F AND 0140 DURING
  172   009D          ; JSR'S, PHP'S, PHA'S AND INTERRUPTS AS NECESSARY.
  173   009D          ; THIS SHOULD BE MORE THAN ENOUGH SPACE FOR THIS PURPOSE.
  174   009D          ;
  175   009D          ; (SEE CODE IN RESET ROUTINE BETWEEN ZLOOP AND FFLOOP ADDRESSES)
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  177   009D              ;
  178   009D              ; THE CODE FOLLOWS. EXECUTION COMES HERE WHEN RESET IS HIT.
  179   009D              ;
  180   009D                        *=$F800
  181   F800  D8         RESET      CLD               DON'T WANT DECIMAL MODE
  182   F801              ;
  183   F801  A2 5B                 LDX #%01011011    RESET NETWORK UART
  184   F803  8E 00 0C              STX NUARTS
  185   F806  A2 D8                 LDX #%11011000    INITIALIZE NETWORK UART
  186   F808  8E 00 0C              STX NUARTS
  187   F80B  A2 40                 LDX #%01000000    TIMER1 IS FREE RUNNING
  188   F80D  8E 13 10              STX P1ACR
  189   F810  A9 00                 LDA #%00000000    PORTS 1 & 3B LINES ARE ALL INPUTS
  190   F812  8D 13 10              STA P1ADDR
  191   F815  8D 12 10              STA P1BDDR
  192   F818  8D 42 10              STA P3BDDR
  193   F81B  8D 23 10              STA P2ACR         DISABLE PORTS 2 & 3 TIMERS AND LATCHING
  194   F81E  8D 43 10              STA P3ACR
  195   F821  A2 96                 LDX #%10011011    PULSE HANDSHAKING (ACTIVE HIGH) FOR PORT 1
  196   F823  8E 1C 10              STX P1PCR
  197   F826  A2 88                 LDX #%10001000    LEVEL HANDSHAKE (ACTIVE LOW) FOR PORT 2
  198   F828  8E 2C 10              STX P2PCR
  199   F82B  A2 F8                 LDX #%11111000    HANDSHAKING LEVEL ON PORT 3A / MANUAL PORT 3B
  200   F82D  8E 4C 10              STX P3PCR
  201   F830  A2 7F                 LDX #%01111111    DISABLE ALL INTERRUPTS FROM PORTS 1,2 & 3
  202   F832  8E 1E 10              STX P1IER
  203   F835  8E 2E 10              STX P2IER
  204   F838  8E 4E 10              STX P3IER
  205   F83B  9A                    TXS               RESET THE STACK POINTER
  206   F83C  A0 82                 LDY #%10000010    ALLOW DISCONNECT INTERRUPT ON PORT 3
  207   F83E  8C 4E 10              STY P3IER
  208   F841  A0 FF                 LDY #%11111111    PORTS 2 & 3A ARE OUTPUTS
  209   F843  8C 23 10              STY P2ADDR
  210   F846  8C 22 10              STY P2BDDR
  211   F849  8C 43 10              STY P3ADDR
  212   F84C  8D 14 10              STA T1MRL         SET TIMER1 TO $4800 (1/100 SECOND)
  213   F84F  A2 48                 LDX #$48
  214   F851  8E 15 10              STX T1MRH         THIS STARTS THE COUNTING
  215   F854  A2 19                 LDX #25           WANT TO COUNT 25 TIMEOUTS (1/4 SECOND)
  216   F856  86 41                 STX TICK
  217   F858              ;
  218   F858  A2 50                 LDX #XADDR-1  ZERO OUT VARIABLES WHICH NEED TO BE RESET -
  219   F85A              ;                           (SEE SPACE ALLOCATION COMMANDS ABOVE)
  220   F85A              ;                           (REG 'A' = 00 FROM ABOVE)
  221   F85A  95 00      ZLOOP      STA 0,X
  222   F85C  CA                    DEX
  223   F85D  10 FB                 BPL ZLOOP
  224   F85F              ;
  225   F85F              ; REG 'Y' = $FF FROM ABOVE
  226   F85F              ;
  227   F85F  A2 02                 LDX #DMATTY-QIN INITIALIZE VARIABLES TO $FF
  228   F861  94 58      FFLOOP     STY QIN,X
  229   F863  CA                    DEX
  230   F864  10 FB                 BPL FFLOOP
  231   F866              ;
```

```
CARD # LOC   CODE      CARD  10        20        30        40        50        60        70
  232  F866  A2 16           LDX #BUFCNT     INITIALLY, BUFCNT ITEMS IN THE STACK
  233  F868  86 36           STX STKPTR      OFFSET OF TOP OF STACK + 1
  234  F86A  CA              DEX             INITIALIZE THE FREE BUFFER STACK
  235  F86B  8A        STACKI TXA
  236  F86C  95 54           STA BUFSTK,X    PUT THE NUMBER OF EACH BUFFER IN THE STACK
  237  F86E  CA              DEX
  238  F86F  10 FA           BPL STACKI
  239  F871                  ;
  240  F871                  ; THE 'A' REG IS SET TO ZERO FROM THE ABOVE LOOP
  241  F871                  ;
  242  F871  A2 15           LDX #BUFCNT-1   INITIALIZE THE FIRST BUFFER POINTER
  243  F873  95 87           STA  HIPTR,X
  244  F875  A9 C0           LDA #<$00C0
  245  F877  95 71           STA LOPTR,X
  246  F879  CA              DEX
  247  F87A  A9 80           LDA #<$0180     SET BUFFER LOW AND HIGH PTRS IN LOPTR & HIPTR
  248  F87C  A0 01           LDY #>$0180     THE SECOND BUFFER IS AT ADDR $0180
  249  F87E  95 71     BUFFRI STA LOPTR,X    SET THE LOW HALF OF THE ADDRESS
  250  F880  98              TYA
  251  F881  95 87           STA HIPTR,X     SET THE HIGH HALF OF THE ADDRESS
  252  F883  B5 71           LDA LOPTR,X
  253  F885  18              CLC             INCR REG 'A' TO POINT TO THE NEXT BUFFER
  254  F886  69 80           ADC #BUFLEN
  255  F888  90 01           BCC SKIPI
  256  F88A  C8              INY             IF THERE WAS A CARRY, INCR THE HIGH HALF
  257  F88B  CA        SKIPI DEX
  258  F88C  10 F0           BPL BUFFRI
  259  F88E                  ;
  260  F88E  AD 40 10        LDA PORT3B      STORE HOME ADDRESS IN RANDOM SEED (********)
  261  F891  85 12           STA RANDU
  262  F893  A9 F0           LDA #240        XMIT FIRST STATUS MSG A MINUTE FROM NOW
  263  F895  85 19           STA TSTAT
  264  F897  A2 FF           LDX #$FF        AND SET CONECT TO WAITING FOR A SYSTEM SIGN-ON
  265  F899                  ;
  266  F899                  ; THE FOLLOWING CODE IS USED TO SEND A SIGN-OFF MESSAGE TO ALL G7 BIUS
  267  F899                  ; TO INSURE THAT A POWER FAILURE WILL CLEAR ANY HUNG CONDITION.  A NUMBER
  268  F899                  ; OF SIGN-OFF MESSAGES EQUIVALENT TO THE NUMBER OF TERMINALS THAT SHOULD
  269  F899                  ; MAKE UP THE CAPS SYSTEM, TRMCNT, IS QUEUED FOR OUTPUT.  IF THIS IS NOT
  270  F899                  ; A POWER-UP CYCLE, ONLY THE CONECT STATE IS CHECKED TO DETERMINE IF
  271  F899                  ; A SIGN-OFF IS REQUIRED.
  272  F899                  ;
  273  F899  A5 3F           LDA WARGO       TEST THE POWER-UP FLAG, IF RESET, = $57
  274  F89B  C9 57           CMP #$57
  275  F89D  F0 29           BEQ NOTPWR      RELIES ON ARBITRARY VALUE DURING POWER-UP
  276  F89F  86 40           STX CONECT      SET CONECT FLAG TO WAITING FOR SIGN-ON (-1)
  277  F8A1  A9 51           LDA #TRMBAS     SET THE INITIAL TERMINAL NUMBER
  278  F8A3  85 3E           STA XADDR
  279  F8A5                  ;
  280  F8A5  A9 0E     SNDSOF LDA #SOFMSG    SIGN OFF MESSAGE TYPE
  281  F8A7  20 AA FD        JSR PCONST      CONSTRUCT THE PACKET
  282  F8AA  10 03           BPL CONSOF
  283  F8AC  4C 00 F4        JMP RESET       IF COULDN'T ALLOCATE A BUFFER, SOMETHING WRONG!
  284  F8AF  A9 17     CONSOF LDA #OBJT-NEXT  ENQUEUE THE PACKET TO BE SENT
  285  F8B1  A4 0E           LDY CURBIU
  286  F8B3  20 14 FE        JSR ENQ
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  287   F8B6              ;
  288   F8B6              ; THIS CODE LOOPS SENDING THE SIGN-OFF MESSAGES UNTIL ALL TERMINALS
  289   F8B6              ;  HAVE BEEN ADDRESSED.
  290   F8B6              ;
  291   F8B6  A9 57               LDA #$57
  292   F8B8  C5 3F               CMP HARGO
  293   F8BA  F0 16               BEQ NOSOF
  294   F8BC  E6 3E               INC XADDR       SET NEXT ADDRESS
  295   F8BE  A6 3E               LDX XADDR       TEST TO SEE IF DONE
  296   F8C0  E0 59               CPX #TRMBAS+TRMCNT
  297   F8C2  D0 E1               BNE SNDSOF      NOT DONE, SO SEND ANOTHER SIGN-OFF MESSAGE
  298   F8C4  85 3F               STA HARGO       DONE, SO FLAG END OF POWER-UP CYCLE
  299   F8C6  D0 0A               BNE NOSOF       ALWAYS BRANCH OUT
  300   F8C8              ;
  301   F8C8              ; TEST FOR A RESET SIGN-OFF MESSAGE
  302   F8C8              ;
  303   F8C8              ;
  304   F8C8  A5 40       NOTPWR  LDA CONECT      GET OLD CONECT FLAG
  305   F8CA  86 40               STX CONECT
  306   F8CC              ;
  307   F8CC              ; TEST TO SEE IF WE ARE ALREADY CONNECTED AND IF SO SEND A SIGN-OFF
  308   F8CC              ;
  309   F8CC  C9 01               CMP #01         WERE WE PREVIOUSLY CONNECTED?
  310   F8CE  D0 02               BNE NOSOF       IF NOT CONNECTED, DON'T SEND SIGN OFF MSG
  311   F8D0  F0 03               BEQ SNDSOF
  312   F8D2              ;*****************************************************************************
  313   F8D2              ;***                                                                       ***
  314   F8D2              ;***  SET UP TO RESPOND TO VOLLEY TEST PROGRAM.                            ***
  315   F8D2              ;***                                                                       ***
  316   F8D2              ;*****************************************************************************
  317   F8D2  A2 B8       NOSOF   LDX #$B8        PREPARE TO SIGN ONTO THE BACKBOARD
  318   F8D4  86 3E               STX XADDR       INDICATE WE ARE TALKING TO BACKBOARD
  319   F8D6  A9 08               LDA #8          SET SIGN-ON TIMER
  320   F8D8  A2 1E               LDX #TSACK
  321   F8DA  20 53 FD            JSR STIMER
  322   F8DD  A9 7F               LDA #$7F        SET THE MAXIMUM PACKET LENGTH
  323   F8DF  85 17               STA MAXPAK
  324   F8E1  A9 59               LDA #SOVMSG     PREPARE SIGN-ON MESSAGE FOR BACKBOARD
  325   F8E3  20 3A FD            JSR PCONST
  326   F8E6  10 03               BPL CONBB
  327   F8E8  4C 00 F8            JMP RESET       IF COULDN'T ALLOCATE A BUFFER, SOMETHING WRONG!
  328   F8EB  A9 17       CONBB   LDA #OUT+NEXT   QUEUE THE SIGN-ON REQUEST
  329   F8ED  A4 0E               LDY CURBIU
  330   F8EF  20 14 FE            JSR ENQ
  331   F8F2  E6 3A               INC WAIT        INDICATE THAT WE ARE WAITING FOR A SIGN-ON ACK
  332   F8F4              ;*****************************************************************************
  333   F8F4              ;***                                                                       ***
  334   F8F4              ;***  SET UP THE READ/WRITE HANDSHAKE LINE.                                ***
  335   F8F4              ;***                                                                       ***
  336   F8F4              ;*****************************************************************************
  337   F8F4  A0 0F               LDY #HARADW     ASSUME HARRIS LAST WROTE TO BIU
  338   F8F6  9C 41 10            STY PORT3A      SIGNAL WRITE INTERRUPT
  339   F8F9  20 44 FC            JSR INTBUF      SET UP A NETWORK INPUT BUFFER
  340   F8FC  58                  CLI             NOW READY TO PROCESS INTERRUPTS
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  342   F8FD              ;
  343   F8FD              ; MLOOP IS THE MAIN LOOP. IT REPEATEDLY CALLS THE VARIOUS PROCESSING
  344   F8FD              ; ROUTINES UNTIL EITHER RESET IS HIT BY THE USER OR A NETWORK/DISCONNECT
▸ 345   F8FD              ; INTERRUPT IS DETECTED.
  346   F8FD              ;
  347   F8FD  20 12 F9    MLOOP   JSR NET         HANDLE A MESSAGE TO THE NETWORK
  348   F900  20 9E FA            JSR OUTDEV      HANDLE A NETWORK MESSAGE FOR THE HARRIS
  349   F903  20 44 FC            JSR INTBUF      SEE IF A NETWORK INPUT BUFFER IS NECESSARY
  350   F906  20 F5 F9            JSR INDEV       HANDLE A HARRIS MESSAGE FOR THE NETWORK
  351   F909  20 6F FB            JSR TIMOUT      POLL THE TIMER
  352   F90C  20 8A FB            JSR CKTOUT      CHECK FOR ANY TIMEOUTS
  353   F90F  4C F5 F8            JMP MLOOP       CONTINUE LOOPING
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  355   F912              ;
  356   F912              ; NET IS CALLED TO OUTPUT MESSAGES ONTO THE NETWORK. IF IT
  357   F912              ; CAN'T FOR ANY REASON, IT RETURNS AND TRIES AGAIN THE NEXT TIME
  358   F912              ; IT IS CALLED.
  359   F912              ; GRAPHIC 7 AND HARRIS VERSIONS OF NET SHOULD BE IDENTICAL.
  360   F912              ;
  361   F912  A6 59       NET       LDX OOUT        IS THERE ANYTHING WAITING TO GO OUT?
  362   F914  30 4D                 BMI NRET        IF NOTHING, RETURN
  363   F916              ;
  364   F916              ; A CHECK IS MADE TO SEE IF WE TRIED TO TRANSMIT SOMETHING ON THE NETWORK
  365   F916              ; EARLIER AND COULD NOT.  THERE ARE TWO POSSIBLE REASONS FOR THIS, A
  366   F916              ; COLLISION WITH SOMEONE ELSE'S TRANSMISSION OR AN 'FF' ACK FROM THE RECEIVER
  367   F916              ; INDICATING HIS BUFFERS ARE FULL.
  368   F916              ;
  369   F916              ; IF A COLLISION OCCURRED, THEN A RANDOM NUMBER IS LOADED INTO RNDCNT
  370   F916              ;
  371   F916              ; IF AN 'FF' ACK WAS RECEIVED, THEN 255 WAS LOADED INTO NOXMIT.
  372   F916              ;
  373   F916              ; THE FOLLOWING LOOPS ALLOW BOTH PROBLEMS TO CLEAR.
  374   F916              ;
  375   F916  A5 37       LWAIT     LDA NOXMIT      WERE WE ASKED TO STOP XMITTING FOR A WHILE?
  376   F918  F0 06                 BEQ SWAIT1      NO, SO CHECK FOR RANDOM BACKOFF
  377   F91A  A9 32                 LDA #50         THIS GIVES US A 0.2 MILLISECOND WAIT
  378   F91C  C6 37                 DEC NOXMIT      MARK THAT WE ARE WAITING 0.2 MS
  379   F91E  85 15                 STA RNDCNT      STORE THE CONSTANT FOR 0.2 MS
  380   F920  A5 15       SWAIT1    LDA RNDCNT      IS RANDOM BACKOFF OR LONG WAITING ACTIVE?
  381   F922  F0 06                 BEQ PREPAR      NO
  382   F924  C6 15       SWAIT2    DEC RNDCNT      COUNTDOWN RANDOM WAIT OR 0.2 MS WAIT
  383   F926  D0 FC                 BNE SWAIT2
  384   F928  F0 EC                 BEQ LWAIT       SEE IF WE STILL HAVE THE LONG WAIT ACTIVE
  385   F92A              ;
  386   F92A              ; PREPARE TO SEND PACKET BY CHECKING TO SEE IF THE NETWORK IS BUSY.  BUILD A
  387   F92A              ; PACKET HEADER WHILE WAITING
  388   F92A              ;
  389   F92A  85 31       PREPAR    STA PARITY      ZERO OUT THE VERTICAL
  390   F92C  A9 FF                 LDA #$FF        SET INTERRUPT DETECTOR
  391   F92E  85 39                 STA INTFLG
  392   F930  AD 00 0C              LDA NUARTS      IS THE NETWORK BUSY?
  393   F933  29 04                 AND #%00000100
  394   F935  F0 27                 BEQ NETBSY      IF BUSY, THEN WAIT FOR LATER
  395   F937              ;
  396   F937              ; NET NOT BUSY NOW BUT COULD BE WAITING FOR AN ACK.  SET UP HEADER AND TEST
  397   F937              ; AGAIN IN 100 USECS.
  398   F937              ;
  399   F937  B4 71                 LDY LOPTR,X     GET THE POINTER TO THE BUFFER TO SEND
  400   F939  84 05                 STY OUTPTR
  401   F93B  B4 87                 LDY HIPTR,X
  402   F93D  84 07                 STY OUTPTR+1
  403   F93F  A5 33                 LDA SNUM        SAVE THE SEQUENCE # IN THE PACKET
  404   F941  A0 04                 LDY #04
  405   F943  91 05                 STA (OUTPTR),Y
  406   F945  A5 32                 LDA NTRX        SAVE THE RETRANSMISSION COUNT IN THE PACKET
  407   F947  A0 06                 LDY #06
  408   F949  91 05                 STA (OUTPTR),Y
  409   F94B  C8                    INY
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
  410   F94C  B1 06             LDA (OUTPTR),Y  GET THE PACKET LENGTH
  411   F94E  85 08             STA OUTBC       SAVE IT FOR  THE OUTPUT LOOP
  412   F950  A0 19             LDY #25         FINISH WAITING FOR 100 US AFTER PREPAR
  413   F952  88          WAIT00 DEY
  414   F953  10 FD             BPL WAIT00
  415   F955  A5 39             LDA INTFLG      GET ZERO, IF INTERRUPT OCCURRED
  416   F957  2D 00 0C          AND NUARTS      GET CURRENT NETWORK STATUS
  417   F95A  29 04             AND #%00000100
  418   F95C  D0 06             BNE TRNSMT      IF IT IS NOT BUSY, OUTPUT A MESSAGE
  419   F95E  A2 2F       NETBSY LDX #NMWAIT    INCR # OF TIMES WE HAVE TO WAIT FOR THE NET.
  420   F960  20 FD FD           JSR SFINC
  421   F963  60          NRET   RTS
  422   F964                     ;
  423   F964                     ; TRANSMIT THE MESSAGE. TURN ON THE XMIT KEY.  THIS SHOULD HOLD THE NET
  424   F964                     ; FOR US, BUT KEEP CHECKING IT TO DETERMINE IF A COLLISION OCCURRED.
  425   F964                     ;
  426   F964  A9 98       TRNSMT LDA #%10011000  TURN ON XMIT KEY
  427   F966  8D 00 0C          STA NUARTS
  428   F969  85 14              STA XMIT        SET XMIT FLAG TO NONZERO VALUE
  429   F96B                     ;
  430   F96B                     ; REG 'Y' = $FF FROM THE WAIT00 LOOP.
  431   F96B                     ;
  432   F96B  C8          TX_LOP INY
  433   F96C  A5 14       TXBEMP LDA XMIT         HAS XMIT FLAG BEEN RESET ?
  434   F96E  D0 06              BNE NTXCOL      NO, SO CONTINUE
  435   F970                     ;
  436   F970                     ; A COLLISION HAS OCCURRED
  437   F970                     ;
  438   F970  A2 25       TXCOL  LDX #NMCOL      INCREMENT THE NUMBER OF COLLISIONS
  439   F972  20 FD FD           JSR SFINC
  440   F975  60                 RTS
  441   F976                     ;
  442   F976                     ; NO COLLISION OCCURRED
  443   F976                     ;
  444   F976  AD 00 0C    NTXCOL LDA NUARTS      IS XMIT BUFFER EMPTY?
  445   F979  29 02              AND #%00000010
  446   F97B  F0 EF              BEQ TXBEMP
  447   F97D  B1 06              LDA (OUTPTR),Y  WRITE WORD TO NETWORK
  448   F97F  8D 01 0C           STA NUARTD
  449   F982  45 31              EOR PARITY      ACCUMULATE THE PARITY
  450   F984  85 31              STA PARITY
  451   F986  C4 08              CPY OUTBC       ARE WE DONE?
  452   F988  D0 E1              BNE TXLOOP      NO, SO LOOP SOME MORE
  453   F98A                     ;
  454   F98A                     ; SEND THE VERTICAL PARITY BYTE
  455   F98A                     ;
  456   F98A  A5 14       TXPAR  LDA XMIT        HAS XMIT BEEN TURNED OFF BY NMI?
  457   F98C  F0 52              BEQ TXCOL       IF SO, RETURN
  458   F98E  AD 00 0C           LDA NUARTS      OUTPUT THE PARITY TO THE NETWORK
  459   F991  29 02              AND #%00000010
  460   F993  F0 F5              BEQ TXPAR
  461   F995  A5 31              LDA PARITY
  462   F997  8D 01 0C           STA NUARTD
  463   F99A  A2 1A              LDX #26         DELAY TURNING OFF THE KEY
  464   F99C  CA          HOLDTX DEX
```

```
CARD # LOC   CODE       CARD  10        20        30        40        50        60        70
  465  F99D  10 FD            BPL HOLDTX
  466  F99F  A9 58            LDA #%01011000  TURN OFF XMIT KEY (THIS MUST OCCUR...
  467  F9A1  8D 00 0C         STA NUARTS      ...AT LEAST 75 USEC AFTER LAST BYTE)
  468  F9A4          ;
  469  F9A4  A5 14    KEYNOF   LDA XMIT        WAIT FOR THE KEY TO TURN OFF
  470  F9A6  D0 FC            BNE KEYNOF
  471  F9A8  78               SEI             NO INTERRUPTS FOR A WHILE
  472  F9A9          ;
  473  F9A9          ; REG 'A' = 00 FROM KEYNOF LOOP ABOVE
  474  F9A9          ;
  475  F9A9  AB               TAY
  476  F9AA  B1 06            LDA (OUTPTR),Y  GET THE TO ADDRESS
  477  F9AC  85 08            STA OUTBC       SAVE IT IN OUTBC
  478  F9AE  E6 14            INC XMIT        MAKE XMIT NON-ZERO
  479  F9B0  A2 11            LDX #17         LOOP FOR 100 US
  480  F9B2  AD 00 0C AKLOOP  LDA NUARTS      WAIT FOR THE ACK TO COME BACK
  481  F9B5  30 1A            BMI ACKCK       A CHARACTER HAS ARRIVED
  482  F9B7  CA               DEX             KEEP ON WAITING
  483  F9B8  10 F8            BPL AKLOOP
  484  F9BA          ;
  485  F9BA          ; 'Y' REG = 00 FROM KEYNOF ABOVE
  486  F9BA          ;
  487  F9BA  84 14    BADACK   STY XMIT        NOTHING ARRIVED, SO TURN OFF XMIT FLAG
  488  F9BC  58               CLI             RESTORE INTERRUPTS
  489  F9BD  A2 23            LDX #NMRXMT     COUNT # OF TIMES NO ACK
  490  F9BF  20 FD FD         JSR SFINC
  491  F9C2  E6 32            INC NTPRX       INCR # OF TIMES THIS MSG IS TRANSMITTED
  492  F9C4  A6 32            LDX NTPRX       SENT 127 TIMES? (LIMIT UPPED CHECKOUT)
  493  F9C6  E0 7F            CPX #127
  494  F9C8  90 23            BCC NETRET      NO, SO TRY IT AGAIN LATER
  495  F9CA  A2 27            LDX #NMDISC     YES, SO INCR THE # OF DISCARDED MESSAGES
  496  F9CC  20 FD FD         JSR SFINC
  497  F9CF  D0 15            BNE NETFRE      AND FREE UP THE PACKET BUFFER
  498  F9D1          ;
  499  F9D1          ; CHECK THE ACKNOWLEDGMENT THAT WAS JUST RECEIVED AND TAKE THE
  500  F9D1          ;   PROPER ACTION.
  501  F9D1          ;
  502  F9D1  AD 01 0C ACKCK    LDA NUARTD      GET THE CHARACTER THAT ARRIVED
  503  F9D4  C5 08            CMP OUTBC       IS IT THE ACK?
  504  F9D6  F0 08            BEQ ACKOK       YES
  505  F9D8  C9 FF            CMP #$FF        IS IT A FLAG THAT THE RECEIVER IS OVERRUN?
  506  F9DA  D0 DF            BNE BADACK      NO
  507  F9DC  85 37            STA NUXMIT      YES, SO DON'T XMIT FOR 255 MILLISECONDS
  508  F9DE  F0 DA            BEQ BADACK      ALWAYS BRANCH
  509  F9E0          ;
  510  F9E0  58       ACKOK    CLI             RESTORE INTERRUPTS
  511  F9E1  A2 21            LDX #NMXMT      INCR THE # OF TRANSMITTED MSGS
  512  F9E3  20 FD FD         JSR SFINC
  513  F9E6          ;
  514  F9E6          ; FREE THE BUFFER THAT WAS JUST PROCESSED OR COULDN'T BE SENT.
  515  F9E6          ;
  516  F9E6  A2 17    NETFRE   LDX #OOUT-NEXT  FREE THE BUFFER THAT WAS SENT
  517  F9E8  20 F2 FE         JSR DQ
  518  F9EB  A9 00            LDA #00         RE-INITIALIZE # OF TIMES CURRENT MSG XMITTED
  519  F9ED  85 32            STA NTPRX
```

```
CARD #  LOC   CODE      CARD  10        20        30        40        50        60        70
  520   F9EF  E6 35           INC SUNUM
  521   F9F1  A5 14   ANDONE  LDA XMIT
  522   F9F3  D0 FC           BNE ANDONE
  523   F9F5  60      NETRET  RTS

                      SET THE SEQUENCE # FOR THE NEXT PACKET
                      WAIT FOR THE KEY TO TURN OFF

                      AND RETURN
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  525   F9F6              ;
  526   F9F6              ; INDEV IS CALLED TO POLL FOR CHARACTERS FROM THE HARRIS.
  527   F9F6              ; TWO STATES OF THE DMATTY FLAG ARE VALID: -1 INDICATING NO OPERATION IS IN
  528   F9F6              ;  PROGRESS AND +1 MEANING A DMA OPERATION IS IN PROGRESS.  UNLIKE THE G7
  529   F9F6              ;  VERSION OF INDEV, ONLY DMA OPERATIONS ARE RECOGNIZED BY THE HARRIS VERSION
  530   F9F6              ;  OF INDEV.
  531   F9F6              ;
  532   F9F6              ;
  533   F9F6              ;    DMA MODE MEANS BINARY DATA IS USED AND CONTROL CHARACTERS MAY BE
  534   F9F6              ;    EMBEDDED, THE FINAL PACKET SHOULD BE TRANSMITTED WHEN SUFFICIENT TIME
  535   F9F6              ;    HAS ELAPSED TO INDICATE NO MORE DATA FOLLOWS (APPROX 600 USECS).
  536   F9F6              ;
  537   F9F6  A5 40       INDEV   LDA CONECT      ARE WE CONNECTED TO ANOTHER BIU?
  538   F9F8  30 09               BMI IRET        NO, SO CONTINUE TO WAIT.
  539   F9FA  A5 0F               LDA INSET       DO WE HAVE AN INPUT BUFFER SET UP YET?
  540   F9FC  D0 1A               BNE GETCH       YES, SO TRY TO GET A CHAR
  541   F9FE  A9 02               LDA #DATMSG+TTY TRY TO ALLOCATE A DATA BUFFER
  542   FA00  20 8A FD           JSR PCONST
  543   FA03  10 01               BPL ABUF        IF COULD ALLOCATE THE BUFFER
  544   FA05  60          IRET    RTS             IF COULDN'T, RETURN AND TRY LATER
  545   FA06              ;
  546   FA06              ; SET UP THE BUFFER POINTERS
  547   FA06              ;
  548   FA06  A4 0A       ABUF    LDY BIUPTR      SET UP THE INPUT BUFFER POINTER
  549   FA08  84 03               STY INPTR
  550   FA0A  A4 0B               LDY BIUPTR+1
  551   FA0C  84 04               STY INPTR+1
  552   FA0E  A4 0E               LDY CURBIU      THE BUFFER NUMBER
  553   FA10  84 0D               STY CURDEV
  554   FA12  E6 0F               INC INSET       NOW SET UP FOR INPUT
  555   FA14  A0 08               LDY #08         OFFSET INTO THE BUFFER OF THE FIRST CHAR
  556   FA16  84 05               STY INBC
  557   FA18              ;
  558   FA18              ; GETCH TRIES TO GET A WORD. IF IT IS SUCCESSFUL, THE WORD IS ADDED TO
  559   FA18              ; THE CURRENT BUFFER.  IF ONLY ONE CHARACTER IS PASSED, IT IS PADDED WITH
  560   FA18              ; A NULL HEX 00.
  561   FA18              ;
  562   FA18  A4 05       GETCH   LDY INBC        SET Y INCASE OF NEED TO OUTPUT BUFFER
  563   FA1A  A9 10               LDA #CDH        IS THERE COMMAND DATA FROM THE HARRIS?
  564   FA1C  2C 13 10            BIT PIIFR
  565   FA1F  D0 5C               BNE GETCDA      YES, SO PROCESS A COMMAND WORD
  566   FA21  A9 02               LDA #ODH        NO, IS A DATA WORD READY?
  567   FA23  2C 13 10            BIT PIIFR
  568   FA26  D0 0A               BNE GETDAT      YES, SO GET THE WORD
  569   FA28              ;
  570   FA28              ; DMATTY IS INITIALIZED TO -1 IN THE FFLOOP.  IT IS SET TO 1 ON THE FIRST
  571   FA28              ; PASS THROUGH GETDAT TO ASSUME A DMA OPERATION.  IT IS SET TO -1 IF THE TIMER
  572   FA28              ; INDICATES IT IS NOT A DMA OPERATION OR A ONE WORD DMA OP AND PACKETS ARE SENT
  573   FA28              ; BY CODE IN IODONE.
  574   FA28              ;
  575   FA28  A5 5A               LDA DMATTY      NO, ARE WE IN THE MIDDLE OF AN OPERATION?
  576   FA2A  30 D9               BMI IRET        NO, SO RETURN
  577   FA2C              ;
  578   FA2C              ; DMATM IS A TIMER SET DURING DMA OPS TO TIME HOW LONG IT HAS BEEN SINCE THE LAS
  579   FA2C              ; WORD TO ARRIVE.  IT IS SET EACH PASS THROUGH GETDAT AND GETCDA
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  580   FA2C              ;
  581   FA2C  C6 3C               DEC DMATM       DMA OPERATION; HAS TIMER GONE OFF?
  582   FA2E  D0 E8               BNE GETCH       NO, SO CONTINUE TO WAIT
  583   FA30              ;
  584   FA30              ; TIMER WENT TO ZERO SO DMA OR TTY OP FINISHED, SO SEND BUFFERRED DATA.
  585   FA30              ;
  586   FA30  F0 52               BEQ INDONE      YES, SO SEND THE PACKET
  587   FA32              ;
  588   FA32  A6 5A       GETDAT  LDX DMATTY      WHAT MODE OF OPERATION IS IN PROGRESS?
  589   FA34  10 04               BPL DMAMOD      DMA
  590   FA36              ;
  591   FA36              ; IF DMATTY = -1, WE ASSUME THIS IS THE START OF A DMA OPERATION.
  592   FA36              ;
  593   FA36  A2 01       DMAIT   LDX #1
  594   FA38  86 5A               STX DMATTY
  595   FA3A  A2 20       DMAMOD  LDX #$20          SET THE DMA TIMER
  596   FA3C  86 3C               STX DMATM
  597   FA3E  AA                  TAX               SAVE THE COMMAND TYPE FOR LATER
  598   FA3F              ;
  599   FA3F  AD 10 10            LDA PORT1B        GET THE HIGH ORDER BYTE
  600   FA42  49 FF               EOR #$FF          TAKE THE 1'S COMPLIMENT OF IT
  601   FA44  48                  PHA               AND SAVE THE BYTE FOR LATER
  602   FA45              ;
  603   FA45              ; NOTE: LOADING FROM PORT 1A WILL TRIGGER AN AUTOMATIC HARDWARE HANDSHAKE
  604   FA45              ;       SIGNAL FROM THE 6522
  605   FA45              ;       OCACP (G7 OCTL) SIGNAL.
  606   FA45              ;
  607   FA45  AD 11 10            LDA PORT1A        GET THE LOW ORDER BYTE
  608   FA48  49 FF               EOR #$FF          TAKE THE 1'S COMPLIMENT OF IT
  609   FA4A  85 33               STA BYTLOW        AND SAVE IT FOR LATER
  610   FA4C  A2 08               LDX #IIFU         TRIGGER TTY OUTPUT INTERRUPT
  611   FA4E  A9 0F               LDA #MARADW       SIGNAL LAST IIFU WAS FOR A WRITE
  612   FA50  8D 41 10            STA PORT3A
  613   FA53              ;
  614   FA53              ; AN OUTPUT INTERRUPT SIGNAL IS SENT BACK TO THE HARRIS FOR SPECIAL
  615   FA53              ; HANDLING.  THIS IS TRIGGERED BY A MANUAL HANDSHAKE ON PORT 3B.
  616   FA53              ;
  617   FA53  8E 4C 10            STX P3PCR
  618   FA56  A2 02               LDX #2
  619   FA58  CA          GETLOP  DEX               LOOP FOR 10 USECS TO GIVE HARRIS TIME TO PROCESS IT
  620   FA59  D0 FD               BNE GETLOP
  621   FA5B  A2 F8               LDX #NIIFU        TURN OFF IIFU HANDSHAKING LINE
  622   FA5D  8E 4C 10            STX P3PCR
  623   FA60  68                  PLA               RESTORE THE HIGH ORDER BYTE
  624   FA61  C0 02               CPY #$04          WAS THIS AN OUTPUT OR COMMAND WORD?
  625   FA63  D0 83               BNE GETCH         COMMAND, SO GO GET ANOTHER WORD
  626   FA65              ;
  627   FA65              ; ADD THE CHARACTER TO THE OUTPUT BUFFER AND QUEUE THE BUFFER WHEN
  628   FA65              ; IT IS FULL.
  629   FA65              ;
  630   FA65  A4 05       CHARS   LDY INBC          THE PLACE TO ADD THE WORD TO THE BUFFER
  631   FA67  91 03               STA (INPTR),Y     STORE IT IN THE BUFFER
  632   FA69  C4 17               CPY MAXPAK        IS THE BUFFER FULL?
  633   FA6B  F0 1C               BEQ INDON1        YES
  634   FA6D  E6 05               INC INBC
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  635   FA6F  20 76 FA           JSR BYTES       DETERMINE WHICH BYTE IT WAS
  636   FA72  B0 F1              BCS CHARS       FIRST, SO GET SECOND
  637   FA74  90 A2              BCC GETCH       SECOND, SO GO GET ANOTHER WORD
  638   FA76                ;
  639   FA76                ; BYTES IS USED TO DETERMIVE WHICH HALF OF A WORD WAS PROCESSED.  THE CARRY
  640   FA76                ; BIT IS USED TO INDICATE AN ODD OR EVEN BYTE COUNT. (ODD=>CS, EVEN=>CC)
  641   FA76                ;
  642   FA76  A5 05         BYTES   LDA INBC        FIRST BYTE IS INDICATED BY AN ODD COUNT
  643   FA78  18                    CLC
  644   FA79  6A                    ROR A           ROTATE THE LOW ORDER BIT INTO THE CARRY
  645   FA7A  A5 34                 LDA BYTLOW      SET UP THE SECOND BYTE IN A REG
  646   FA7C  60            BYTRET  RTS
  647   FA7D                ;
  648   FA7D.               ; PROCESS A COMMAND DATA WORD (I.E. IGNORE IT BUT HANDSHAKE)
  649   FA7D                ;
  650   FA7D  A6 5A         GETCDA  LDX DMATTY      IF PREVIOUSLY IN TTY OR DMA CLEAR THE BUFFERS
  651   FA7F  10 03                 BPL INDONE      YES
  652   FA81  AB                    TAY             SAVE THE COMMAND TYPE FOR LATER
  653   FA82  D0 B2                 BNE DMAIT       OTHERWISE, START A DMA OPERATION (ALWAYS.BRANCH)
  654   FA84                ;
  655   FA84                ; INDONE IS USED TO SEND PACKETS THAT HAVE BEEN PREVIOUSLY BUFFERED.
  656   FA84                ; IT CHECKS FOR NULL PACKETS (INBC<8) AND RESETS THE TIMERS
  657   FA84                ;
  658   FA84  A2 FF         INDONE  LDX #$FF        RESET DMATTY FLAG
  659   FA86  86 5A                 STX DMATTY
  660   FA88  88                    DEY             SET PACKET LENGTH
  661   FA89  98            INDON1  TYA             PLACE THE BYTE COUNT IN THE PACKET
  662   FA8A  C0 07                 CPY #7          IS IT A NULL PACKET?
  663   FA8C  F0 EE                 BEQ BYTRET      YES, SO IGNORE IT
  664   FA8E  A0 07                 LDY #07
  665   FA90  91 03                 STA (INPTR),Y
  666   FA92  A9 17                 LDA #BOUT+NEXT  ENQUEUE THE PACKET TO BE SENT
  667   FA94  A4 00                 LDY CURDEV
  668   FA96  20 14 FE              JSR ENQ
  669   FA99  C6 0F                 DEC INSET       NO LONGER SET UP
  670   FA9B  4C F6 F9              JMP INDEV       TRY TO GET THE NEXT BUFFER
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  672   FA9E              ;
  673   FA9E              ; OUTDEV IS CALLED TO HANDLE A MESSAGE FROM THE NETWORK. IT WILL
  674   FA9E              ; EITHER TRANSMIT THE MESSAGE TO THE HARRIS OR HANDLE THE
  675   FA9E              ; MESSAGE ITSELF, DEPENDING ON THE MESSAGE TYPE.
  676   FA9E              ;
  677   FA9E              ; OUTDEV MAKES NO ASSUMPTION CONCERNING THE TPYE OF OPERATION (I.E.
  678   FA9E              ; DMA OR TTY), IT SIMPLY SENDS DATA INTO THE DEVICE AS IT IS RECEIVED.
  679   FA9E              ;
  680   FA9E  A5 10       OUTDEV  LDA OUTSET      SET UP FOR OUTPUT?
  681   FAA0  D0 27               BNE PUTCH       YES, SO SEE IF THE NEXT CHARACTER CAN BE SENT.
  682   FAA2  A6 58               LDX QIN         ANYTHING ON THE QUEUE?
  683   FAA4  30 D6               BMI BYTRET      NO, SO RETURN
  684   FAA6  B4 71               LDY LOPTR,X     SET UP THE PTRS
  685   FAA8  84 06               STY OUTPTR
  686   FAAA  B4 87               LDY HIPTR,X
  687   FAAC  84 07               STY OUTPTR+1
  688   FAAE  A0 05               LDY #05         GET THE MSG TYPE
  689   FAB0  B1 06               LDA (OUTPTR),Y
  690   FAB2  C9 02               CMP #DATMSG+TTY A DATA MESSAGE?
  691   FAB4  D0 59               BNE TRYSOF      NO, SO TRY FOR A SIGN-OFF MSG
  692   FAB6  A4 40               LDY CONECT      ARE WE CONNECTED?
  693   FAB8  10 05               BPL PDATA       YES, SO PRINT THE DATA
  694   FABA  EA                  NOP             PATCH FOR VOLLEY PROGRAM 2/10/80
  695   FABB  EA                  NOP
  696   FABC  4C 69 F8            JMP OUTFRE      NO, SO IGNORE THE PACKET
  697   FABF              ;
  698   FABF              ; PDATA SENDS THE DATA TO THE HARRIS
  699   FABF              ;
  700   FABF  E6 10       PDATA   INC OUTSET      SET UP FOR OUTPUT
  701   FAC1  A0 07               LDY #07
  702   FAC3  84 08               STY OUTBC       FIRST CHAR OFFSET - 1
  703   FAC5  B1 06               LDA (OUTPTR),Y  GET THE PACKET LENGTH
  704   FAC7  85 09               STA OUTPL
  705   FAC9              ;
  706   FAC9  A4 08       PUTCH   LDY OUTBC       GET THE NEXT CHAR OFFSET
  707   FACB  C4 09               CPY OUTPL       DONE YET?
  708   FACD  D0 05               BNE TRYDS       NO, SO CHECK THE DEVICE STATUS
  709   FACF  C6 10               DEC OUTSET      ALL DONE WITH THE PACKET
  710   FAD1  4C 69 F8            JMP OUTFRE
  711   FAD4              ;
  712   FAD4              ; TRYDS OUTPUTS TWO CHARACTERS AT
  713   FAD4              ; A TIME PADDING WITH A NULL HEX 00 IF AN ODD NUMBER.
  714   FAD4              ;
  715   FAD4              ; HANDSHAKING OCCURS AUTOMATICALLY WHEN DATA IS STORED IN PORT 2A
  716   FAD4              ; AGAIN A HARDWARE ACTION OF THE 6522 CHIP WHICH IS EQUIVALENT TO THE
  717   FAD4              ; DAVFU (G7 IWR) SIGNAL.
  718   FAD4              ;
  719   FAD4  C8          TRYDS   INY             GET THE OFFSET OF THE CHAR TO SEND
  720   FAD5  B1 06               LDA (OUTPTR),Y  GET THE CHAR TO SEND
  721   FAD7  49 FF               EOR #$FF        TAKE THE 1'S COMPLIMENT OF IT
  722   FAD9  AA                  TAX             SAVE THE FIRST BYTE IN X
  723   FADA  C4 09               CPY OUTPL       DONE YET?
  724   FADC  F0 20               BEQ NULL        YES, SO PUT OUT A NULL CHARACTER TO FILL WORD
  725   FADE  C8                  INY             NO, SO GET SECOND BYTE OF WORD
  726   FADF  B1 06               LDA (OUTPTR),Y  GET THE SECOND CHARACTER
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
  727   FAE1  49 FF     NULRET  EOR #$FF         TAKE THE 1'S COMPLIMENT OF IT
  728   FAE3  8E 20 10         STX PORT2B       LOAD B WITH HIGH ORDER BYTE
  729   FAE6  8D 21 10         STA PORT2A       LOAD A WITH THE LOW ORDER BYTE AND HANDSHAKE
  730   FAE9                  ;
  731   FAE9                  ; A MANUAL HANDSHAKE ON PORT 3B IS USED TO ACT AS AN INPUT INTERRUPT.
  732   FAE9                  ;
  733   FAE9  A9 D8             LDA #IIFU      TRIGGER A HARRIS INTERRUPT
  734   FAEB  A2 2F             LDX #HARADR    SIGNAL THIS IIFU IS FOR A READ
  735   FAED  8E 41 10          STX PORT3A
  736   FAF0  8D 4C 10          STA P3PCR
  737   FAF3                  ;
  738   FAF3                  ; OUTLOP WAITS FOR THE HARRIS TO SIGNAL IT HAS READ THE LAST WORD SENT.
  739   FAF3                  ; IT WAITS FOR ABOUT 2.0 MSECS FOR THE RETURN HANDSHAKE.  IF THE HANDSHAKE
  740   FAF3                  ; FAILS TO COME, IT ASSUMES THERE IS AN INTERFACE DEADLOCK AND RETURNS TO
  741   FAF3                  ; THE MAIN LOOP TO ALLOW IT TO CLEAR.
  742   FAF3                  ;
  743   FAF3  A2 E0             LDX #$E0        SET LOOP COUNTER TO CLEAR DEADLOCKS (2.0 MS)
  744   FAF5  CA        OUTLOP  DEX
  745   FAF6  F0 09             BEQ TRYLAT     IF LOOP COUNTER = 0, TRY LATER
  746   FAF8  A9 02             LDA #DATU      TEST FOR HANDSHAKING FROM HARRIS
  747   FAFA  2C 20 10          BIT P2IFR
  748   FAFD  F0 F6             BEQ OUTLOP     LOOP UNTIL HARRIS ACKNOWLEDGES WORD
  749   FAFF  84 08             STY OUTBC      SAVE BYTE COUNT
  750   FB01  A9 F8     TRYLAT  LDA #NIIFU     TURN OFF INPUT INTERRUPT FLAG
  751   FB03  8D 4C 10          STA P3PCR
  752   FB06  E0 00             CPX #00        ARE WE WAITING TO TRY LATER?
  753   FB08  D0 BF             BNE PUTCH      NO, SO GET THE NEXT WORD (ALWAYS BRANCH)
  754   FB0A  60                RTS            YES, SO GO BACK TO MAIN LOOP
  755   FB0B                  ;
  756   FB0B                  ; THESE TWO LINES OF CODE ADD THE NULL BYTE DURING OUTPUT.
  757   FB0B                  ;
  758   FB0B  A9 00     NULL    LDA #$00       LOAD SECOND BYTE WITH NULL CHARACTER
  759   FB0D  F0 D2             BEQ NULRET     ALWAYS BRANCH
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
  760   FB0F             ;
  761   FB0F             ; THE FOLLOWING SECTIONS CHECK FOR SPECIAL MESSAGES
  762   FB0F             ;
  763   FB0F  C9 0E      TRYSOF    CMP #SOFMSG     WAS IT A SIGN-OFF MESSAGE?
  764   FB11  D0 17               BNE TRYSAC      NO, SO TRY FOR SIGN-ON ACKNOWLEDGMENT
  765   FB13  A5 40               LDA CONECT      CONNECTED?
  766   FB15  30 52               BMI OUTFRE      NO, SO IGNORE THE MSG
  767   FB17  A5 3E               LDA XADDR        IS THE PACKET FROM THE BIU TO WHICH CONNECTED?
  768   FB19  A0 02               LDY #2
  769   FB1B  D1 06               CMP (OUTPTR),Y
  770   FB1D  D0 4A               BNE OUTFRE       NO, SO IGNORE THE SIGN-OFF
  771   FB1F  A0 00               LDY #00         CLEAR XADDR, INBC, AND CONECT
  772   FB21  84 3E               STY XADDR
  773   FB23  84 05               STY INBC
  774   FB25  88                  DEY
  775   FB26  84 40               STY CONECT
  776   FB28  D0 3F               BNE OUTFRE      ALL DONE WITH THE MSG (ALWAYS BRANCH)
  777   FB2A             ;
  778   FB2A  C9 0F      TRYSAC    CMP #SACMSG     WAS IT A SIGN-ON ACK?
  779   FB2C  D0 0B               BNE TRYSON      NO, SO TRY FOR SIGN-ON REQUEST
  780   FB2E  A4 3A               LDY WAIT        YES, SO CHECK TO MAKE SURE WE ARE WAITING
  781   FB30  F0 37               BEQ OUTFRE      NO, SO IGNORE THE PACKET
  782   FB32  88                  DEY
  783   FB33  84 3A               STY WAIT        NO LONGER WAITING FOR SIGN-ON ACK
  784   FB35  84 40               STY CONECT      SET UP TO RUN VOLLEY
  785   FB37  F0 30               BEQ OUTFRE      NOW CLEAR THE PACKET
  786   FB39             ;
  787   FB39  C9 E0      TRYSON    CMP #SONMSG     SIGN-ON REQUEST?
  788   FB3B  90 2C               BCC OUTFRE      NO, SO IGNORE THE PACKET
  789   FB3D  A4 40               LDY CONECT      ARE WE CONNECTED?
  790   FB3F  C0 01               CPY #01
  791   FB41  F0 26               BEQ OUTFRE      YES, SO IGNORE THE PACKET
  792   FB43  A0 02               LDY #2          NO, SO GET THE TERMINAL'S ADDRESS
  793   FB45  B1 06               LDA (OUTPTR),Y
  794   FB47  85 3E               STA XADDR       AND SAVE IT
  795   FB49  A0 00               LDY #00         CLEAR THE WAIT FLAG AND PREPARE TO RESET THE
  796   FB4B  84 3A               STY WAIT         ... TO ADDRESS
  797   FB4D             ;
  798   FB4D             ;  ************************************************************
  799   FB4D             ;  **** SPECIAL CODE TO HANDLE VOLLEY TEST PROGRAM FROM ****
  800   FB4D             ;  ************************************************************
  801   FB4D             ;
  802   FB4D  A6 0F               LDX INSET       WERE WE PREVIOUSLY SET UP FOR AN INPUT?
  803   FB4F  F0 02               BEQ CONTYS      NO, SO CONTINUE
  804   FB51  91 03               STA (INPTR),Y   RESET THE TO ADDRESS
  805   FB53             ;
  806   FB53             ;  ************************************************************
  807   FB53             ;
  808   FB53  A9 57      CONTYS    LDA #$57        GOING TO TALK TO TERMINAL
  809   FB55  85 17               STA MAXPAX      SO SAVE MAX NUMBER OF CHARS/PACKET
  810   FB57  A9 0F               LDA #SACMSG     SEND A SIGN-ON ACK
  811   FB59  20 8A FD            JSR PCONST
  812   FB5C  30 03               BMI OUTFRE      IF CAN'T ALLOCATE THE SIGN-ON, IGNORE IT
  813   FB5E  A9 17               LDA #QOUT-NEXT  QUEUE THE MESSAGE TO BE SENT
  814   FB60  A4 0E               LDY CURBIU
```

```
CARD # LOC   CODE       * CARD 10        20        30        40        50        60        70
  815  FB62  20 14 FE          JSR ENQ
  816  FB65  A0 01             LDY #1
  817  FB67  84 40             STY CONECT
  818  FB69                 ;
  819  FB69                 ; FREE THE PACKET JUST PROCESSED
  820  FB69                 ;
  821  FB69  A2 16          OUTFRE  LDX #JIV-NEXT   FREE UP THE PACKET
  822  FB6B  20 22 FE               JSR DJ
  823  FB6E  60                     RTS             AND DONE
```

```
CARD # LOC   CODE        CARD  10        20        30        40        50        60        70
  825  FB6F             ;
  826  FB6F             ; TIMOUT POLLS THE TIMER TO SEE IF IT HAS TIMED OUT. IF IT HAS, IT
  827  FB6F             ; DECREMENTS TICK AND POSSIBLY DECREMENTS TIMER. IF TICK REACHES 0
  828  FB6F             ; (ONCE EACH SECOND) TOD IS INCREMENTED.
  829  FB6F             ;
  830  FB6F 2C 13 10    TIMOUT  BIT PIIER       HAS THE TIMER TIMED OUT?
  831  FB72 50 15               BVC TIMRET      NO
  832  FB74 AD 14 10            LDA TIMRL       YES, SO CLEAR THE FLAG
  833  FB77 C6 41               DEC TICK        COUNT DOWN UNTIL 1/4 SECOND HIT
  834  FB79 D0 0E               BNE TIMRET
  835  FB7B A9 19               LDA #25         RESET TICK TO COUNT NEXT 1/4 SECOND
  836  FB7D 85 41               STA TICK
  837  FB7F E6 18               INC TOD         INCREMENT TIME OF DAY
  838  FB81 D0 06               BNE TIMRET      IF NO CARRY, ALL DONE
  839  FB83 E6 19               INC TOD+1
  840  FB85 D0 02               BNE TIMRET
  841  FB87 E6 1A               INC TOD+2
  842  FB89 60          TIMRET  RTS
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  844   FB8A              ;
  845   FB8A              ; CKTOUT CHECKS FOR ANY TIMEOUTS AND TAKES APPROPRIATE ACTIONS
  846   FB8A              ; IF ANY ARE DETECTED.
  847   FB8A              ;
  848   FB8A              ; SEE IF WE HAVE WAITED FOR A SIGN-ON ACK LONG ENOUGH
  849   FB8A              ;
  850   FB8A  A5 33       CKTOUT  LDA WAIT          ARE WE WAITING FOR A SIGN-ON ACK?
  851   FB8C  F0 10               BEQ CKSTAT        NO, SO CHECK FOR STATUS
  852   FB8E  A2 1E               LDX #ISACK        YES, SO SET UP POINTER
  853   FB90  20 A9 FB            JSR CTIME
  854   FB93  90 09               BCC CKSTAT
  855   FB95  A2 00               LDX #00           WAITED TOO LONG SO CLEAR FLAGS
  856   FB97  86 33               STX WAIT
  857   FB99  86 5E               STX XADDR
  858   FB9B  CA                  DEX
  859   FB9C  86 4D               STX CONECT
  860   FB9E              ;
  861   FB9E              ; SEE IF IT IS TIME TO SEND A STATUS MESSAGE.
  862   FB9E              ;
  863   FB9E  A2 14       CKSTAT  LDX #ISTAT
  864   FBA0  20 A9 FB            JSR CTIME
  865   FBA3  90 03               BCC CKCRET        NOT TIMED OUT, SO RETURN
  866   FBA5  20 4E FB            JSR SNDSTA
  867   FBA8  60          CKCRET  RTS
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
  869   FBA9             ;
  870   FBA9             ; CTIME IS USED TO COMPARE THE CURRENT TIME OF DAY TO THE VALUE
  871   FBA9             ; STORED IN THE VARIABLE X,X+1 AND X+2.(3 BYTES REQUIRED).  IF THE
  872   FBA9             ; TOD IS GREATER THAN OR EQUAL TO THE VARIABLE CARRY WILL BE SET
  873   FBA9             ; UPON RETURN; OTHERWISE CARRY IS CLEARED.
  874   FBA9             ;
  875   FBA9  A5 1A      CTIME   LDA TOD+2
  876   FBAB  D5 02               CMP 2,X
  877   FBAD  90 0E               BCC CTRET
  878   FBAF  D0 0C               BNE CTRET       TOD LESS THAN X
  879   FBB1  A5 19               LDA TOD+1       GET SECOND BYTE
  880   FBB3  D5 01               CMP 1,X
  881   FBB5  90 06               BCC CTRET
  882   FBB7  D0 04               BNE CTRET       TOD LESS THAN X
  883   FBB9  A5 18               LDA TOD
  884   FBBB  D5 00               CMP 0,X
  885   FBBD  60         CTRET    RTS
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
  887   F8BE             ;
  888   F8BE             ; SNDSTA IS USED TO SEND A STATUS MESSAGE
  889   F8BE             ;
  890   F8BE  A9 DB      SNDSTA   LDA #STAMSG     SEND A STATUS MSG
  891   F8C0             ;
  892   F8C0             ; IF THIS HARRIS BIU IS TO BE USED TO GATHER STATUS MESSAGES IT IS NECESSARY
  893   F8C0             ; TO HALT ITS SENDING STATUS MESSAGES ON THE NETWORK.TO DO THIS CODE AT THE
  894   F8C0             ; BOTTOM OF THE ROUTINE IS ALTERED TO ADD THE STATUS MESSAGE DIRECTLY ONTO THE
  895   F8C0             ; INPUT QUEUE.  IF THE HARRIS IS NOT SET UP TO RECEIVE STATUS MESSAGES BUT
  896   F8C0             ; IT IS DESIRABLE TO CAPTURE AND IGNORE THEM (HALT RETRANSMISSIONS) THIS
  897   F8C0             ; BIU MUST NOT GENERATE A STATUS MESSAGE.
  898   F8C0             ;
  899   F8C0             ; 60      RTS
  900   F8C0             ;
  901   F8C0             ; TO HALT THE GENERATION OF STATUS MESSAGES REPLACE THE BELOW NOP
  902   F8C0             ; WITH THE ABOVE RTS
  903   F8C0             ;
  904   F8C0  EA                  NOP
  905   F8C1  20 BA FD            JSR PCONST       TRY TO GET A BUFFER
  906   F8C4  30 E2               BMI CANRET       NOT ALLOCATED, SO TRY LATER
  907   F8C6  A0 00               LDY #00          SET THE TO ADDRESS TO THE DEVICE THAT RECORDS STATUS
  908   F8C8  A9 00               LDA #STATUS
  909   F8CA  91 0A               STA (OUTPTR),Y
  910   F8CC  A9 29               LDA #41          PACKET LENGTH OF 42
  911   F8CE  A0 07               LDY #07
  912   F8D0  91 0A               STA (OUTPTR),Y
  913   F8D2             ;
  914   F8D2  C8                  INY              POINT TO THE FIRST BYTE OF STATUS PART
  915   F8D3  A2 00               LDX #00          MOVE IN THE 16 BYTES OF STATUS INFORMATION
  916   F8D5  B5 F1       SLOOP1  LDA NSTMT,X      GET A BYTE
  917   F8D7  91 0A               STA (OUTPTR),Y   AND SAVE IT IN THE MESSAGE
  918   F8D9  E8                  INX              UPDATE THE POINTERS AND LOOP COUNTER
  919   F8DA  C8                  INY
  920   F8DB  E0 10               CPX #16          SIXTEEN BYTES MOVED YET?
  921   F8DD  90 F5               BCC SLOOP1       BRANCH IF NO
  922   F8DF             ;
  923   F8DF  A9 00               LDA #00          GET THE # OF MSGS WAITING TO GO OUT
  924   F8E1  A6 59               LDX OUT
  925   F8E3  30 0C               BMI SLOT2        IF NONE
  926   F8E5  18          SLOOP2  CLC              INCR # WAITING
  927   F8E6  69 01               ADC #01
  928   F8E8  48                  PHA              SAVE A
  929   F8E9  B5 42               LDA NEXT,X       GET THE PTR TO THE NEXT MSG
  930   F8EB  AA                  TAX
  931   F8EC  68                  PLA              RESTORE A
  932   F8ED  E0 FF               CPX #$FF         IS THE NEXT PTR NULL?
  933   F8EF  D0 F4               BNE SLOOP2       NO, SO GET THE NEXT ONE
  934   F8F1  91 0A       SLOT2   STA (OUTPTR),Y   SAVE THE # OF WAITING MSGS
  935   F8F3             ;
  936   F8F3  A9 00               LDA #00          GET THE # OF MSGS WAITING TO GO IN
  937   F8F5  A6 58               LDX QIN
  938   F8F7  30 0C               BMI SLOT3        IF NONE
  939   F8F9  18          SLOOP3  CLC
  940   F8FA  69 01               ADC #01
  941   F8FC  48                  PHA
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  942   FBFD  B5 42              LDA NEXT,X
  943   FBFF  AA                 TAX
  944   FC00  68                 PLA
  945   FC01  E0 FF              CPX #$FF
  946   FC03  D0 F4              BNE SLOOP3
  947   FC05  C8          SBJT3  INY               SAVE THE # OF WAITING MSGS
  948   FC06  91 0A              STA (BIUPTR),Y
  949   FC08              ;
  950   FC09  C8                 INY
  951   FC09  AD 00 0C           LDA NUARTS        NETWORK UART STATUS
  952   FC0C  91 0A              STA (BIUPTR),Y
  953   FC0E  C8                 INY
  954   FC0F  AD 00 14           LDA DUARTS        DEVICE UART STATUS
  955   FC12  91 0A              STA (BIUPTR),Y
  956   FC14  C8                 INY
  957   FC15  AD 10 10           LDA PIIFR         PARALLEL PORT INTERRUPT FLAG REGISTER
  958   FC18  91 0A              STA (BIUPTR),Y
  959   FC1A  C8                 INY
  960   FC1B  A5 3E              LDA XMTADR        THE CURRENT XMIT ADDRESS
  961   FC1D  91 0A              STA (BIUPTR),Y
  962   FC1F              ;
  963   FC1F  C8                 INY               POINT TO THE FIRST BYTE OF THE DESCRIPTOR
  964   FC20              ;
  965   FC20              ; REG 'X' = -1 FROM LOOP ABOVE
  966   FC20              ;
  967   FC20  E8                 INX               MOVE IN 12 BYTES OF BIU FUNCTION DESCRIPTION
  968   FC21  B0 34 FE    SLOOP4 LDA BIUFNC,X      GET A BYTE
  969   FC24  91 0A              STA (BIUPTR),Y    AND SAVE IT IN THE MESSAGE
  970   FC26  EA                 TXA               UPDATE THE POINTERS AND LOOP COUNTER
  971   FC27  C8                 INY
  972   FC28  E0 0C              CPX #12           TWELVE BYTES MOVED YET?
  973   FC2A  90 F5              BCC SLOOP4        BRANCH IF NO
  974   FC2C              ;
  975   FC2C              ;       LDA #.IN-NEXT   PUT THE STATUS MESSAGE ON THE INPUT QUEUE
  976   FC2C              ; NOTE: THIS IS SPECIAL CODE DESIGNED FOR ONLY ONE OF THE HARRIS BIU'S,
  977   FC2C              ;       THE ONE THAT IS TASKED TO RETREIVE STATUS MESSAGES FROM THE
  978   FC2C              ;       BUS.  FOR ALL OTHER HARRIS BIU'S THE FOLLOWING FORM IS USED.
  979   FC2C              ;
  980   FC2C  A9 17              LDA #.OUT-NEXT    SEND THE STATUS MSG
  981   FC2E              ;
  982   FC2E  A4 0E              LDY CURBIU
  983   FC30  20 14 FE           JSR E..
  984   FC33  A9 F0              LDA #240          THE NEXT STATUS MSG GOES OUT IN 1 MINUTE
  985   FC35  A2 19              LDX #ISTAT
  986   FC37  20 E8 FD           JSR STIMER
  987   FC3A              ;
  988   FC3A  A9 00              LDA #00           CLEAR THE STATUS COUNTERS AND RETURN
  989   FC3C  A2 0F              LDX #15
  990   FC3E  95 21       SLOOP5 STA MSTAT,X
  991   FC40  CA                 DEX
  992   FC41  10 F9              BPL SLOOP5
  993   FC43              ;
  994   FC43  60                 RTS
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  996   FC44              ;
  997   FC44              ; INTBUF SETS UP A BUFFER FOR NETWORK INPUT IF IT IS NECESSARY.
  998   FC44              ;
  999   FC44  08          INTBUF  PHP             PROTECTS THE INTERRUPT MASKING.
 1000   FC45  78                  SEI             DON'T WANT INTERRUPTS DURING ALLOC
 1001   FC46  A5 11               LDA INTSET
 1002   FC48  D0 11               BNE INTRES      ALREADY HAVE A BUFFER, SO RETURN
 1003   FC4A  20 0A FE            JSR ALLOC       ALLOCATE A BUFFER
 1004   FC4D  30 0C               BMI INTRES      NO BUFFERS AVAILABLE, SO TRY AGAIN LATER
 1005   FC4F  B5 71               LDA LOPTR,X     SET UP THE PTR TO THE BUFFER
 1006   FC51  85 00               STA INPTR
 1007   FC53  B5 87               LDA HIPTR,X
 1008   FC55  85 01               STA INPTR+1
 1009   FC57  86 0C               STX CURNET      SAVE THE BUFFER NUMBER
 1010   FC59  E6 11               INC INTSET      NOW ALL SET UP
 1011   FC5B  28          INTRES  PLP             RESTORE OLD INTERRUPT STATUS AND RETURN
 1012   FC5C  60                  RTS
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
 1014   FC5D              ;
 1015   FC5D              ; IRQ RECEIVES CONTROL WHENEVER THERE IS A NETWORK INTERRUPT.  THIS
 1016   FC5D              ; CAN BE GENERATED BY THE RECEIVE DATA REGISTER OF THE NETWORK UART BEING
 1017   FC5D              ; FULL, THE DATA CARRIER DETECT SIGNAL GOING HIGH, OR A DATA OVERRUN OR
 1018   FC5D              ; IT COULD ALSO BE THE RESULT OF A DISC INTERRUPT FROM THE HARRIS.
 1019   FC5D              ;
 1020   FC5D  48          IRQ       PHA             PUSH ACC, Y AND X ONTO STACK
 1021   FC5E  98                    TYA
 1022   FC5F  48                    PHA
 1023   FC60  8A                    TXA
 1024   FC61  48                    PHA
 1025   FC62              ;
 1026   FC62              ; FIRST TEST TO SEE IF THIS IS A DISCONNECT INTERRUPT FROM
 1027   FC62              ;
 1028   FC62  A9 02               LDA #DISC
 1029   FC64  2C 4D 10            BIT PSIFR         IS THE DISC FLAG SET?
 1030   FC67  F0 19               BEQ NINT          NO, SO PROCESS A NETWORK INTERRUPT
 1031   FC69  A9 00               LDA #00           SET UP TO READ ADDRESS INPUT ON PORT 3A
 1032   FC6B  8D 43 10            STA P3ADDR
 1033   FC6E  A9 0F               LDA #$0F          SET UP TO TEST THE UNIT CODE
 1034   FC70  2D 41 10            AND PORT3A        GET THE FOUR LOW ORDER BITS
 1035   FC73  C9 0F               CMP #MARADW       TEST THE ADDRESS.  IS IT FOR THIS BIU?
 1036   FC75  D0 03               BNE DCR0          NO, SO IGNORE THE INTERRUPT
 1037   FC77  AD 41 10            LDA PORT3A        YES, SO HANDSHAKE AND SET THE CNCT FLAG
 1038   FC7A  A9 FF       DCR0    LDA #$FF          RETURN PORT 3A TO OUTPUT STATUS
 1039   FC7C  8D 43 10            STA P3ADDR
 1040   FC7F  4C 45 FD            JMP DCRET         AND ALWAYS BRANCH
 1041   FC82              ;
 1042   FC82              ; THIS CODE IS THE NETWORK UART INTERRUPT HANDLER. IT WILL TRY TO
 1043   FC82              ; CHECK FOR VALIDITY, OR TO READ IN A MSG FROM ANOTHER BIU.  THIS
 1044   FC82              ; VERSION IS SET TO USE THE SWITCH SELECTABLE ADDRESS AT PORT3B AND ACK
 1045   FC82              ; STATUS MESSAGES.
 1046   FC82              ;
 1047   FC82  AC 00 0C    NINT      LDY NUARTS        GET THE STATUS(CLEARS THE OVERRUN CONDITION)
 1048   FC85  AD 01 0C              LDA NUARTD        AND THE DATA
 1049   FC88  84 02                 STY INTBC         SAVE THE PARITY ERROR FLAG
 1050   FC8A  A6 39                 LDX INTX          INTX IS INITIALLY ZERO
 1051   FC8C  A0 14                 LDY XMIT          IS THIS MY TRANSMISSION?
 1052   FC8E  F0 3F                 BEQ RXDATA        IF NOT, GO TO RECEIVE DATA ROUTINE
 1053   FC90  95 15                 STA TOKEN,X       SAVE RECEIVED DATA IN TOKEN
 1054   FC92  E8                    INX
 1055   FC93  E0 02                 CPX #02           HAVE 2 CHARACTERS BEEN RECEIVED?
 1056   FC95  F0 03                 BEQ TESTIX        YES, SO TEST THEM
 1057   FC97  4C 93 FD              JMP NIRET         NO, SO HAVE TO WAIT, BUT KEEP DECREMENTING
 1058   FC9A              ;
 1059   FC9A  A0 01       TESTIX    LDY #01           TEST RCV'D VS. TRANSMITTED DATA
 1060   FC9C  B9 15 00    TEST2     LDA TOKEN,Y
 1061   FC9F  D1 06                 CMP (OUTPTR),Y    OUTPTR POINTS TO ANY PACKET BEING XMITTED
 1062   FCA1  D0 08                 BNE COLIDE
 1063   FCA3  88                    DEY
 1064   FCA4  10 F6                 BPL TEST2
 1065   FCA6  A9 18                 LDA #%00011000    TEST OK, TURN OFF NET RCVR
 1066   FCA8  8D 00 0C              STA NUARTS
 1067   FCAB  4C 93 FD              JMP NICLRX
 1068   FCAE              ;
```

```
CARD #  LOC   CODE        CARD   10        20        30        40        50        60        70
 1069   FCAE  A9 58     COLIDE   LDA #%01011000  TEST FAILED, TURN OFF XMIT AND RCVR
 1070   FCB0  8D 00 0C           STA MUARTS
 1071   FCB3  A5 12              LDA RANDU       CHOOSE RANDOM WAIT
 1072   FCB5  0A                 ASL A           TO GET NEXT RANDOM NUMBER, ...
 1073   FCB6  0A                 ASL A            MULTIPLY BY 13 AND ADD 1
 1074   FCB7  85 02              STA INTBC       X4
 1075   FCB9  0A                 ASL A           X8
 1076   FCBA  18                 CLC
 1077   FCBB  65 02              ADC INTBC       X12
 1078   FCBD  18                 CLC
 1079   FCBE  65 12              ADC RANDU       X13
 1080   FCC0  18                 CLC
 1081   FCC1  69 01              ADC #01         +1
 1082   FCC3  85 12              STA RANDU       STORE NEW RANDOM NUMBER AT SEED AND COUNT
 1083   FCC5  85 13              STA RNDCNT
 1084   FCC7  A2 25              LDX #NMCOL      INCREMENT # OF COLLISIONS
 1085   FCC9  20 F0 FD           JSR SETVC
 1086   FCCC  4C 9B FD           JMP NICLRX
 1087   FCCF                ;
 1088   FCCF  CD 40 10   RXDATA  CMP PORTSB      IS THE PACKET FOR US?
 1089   FCD2  F0 0C              BEQ FORUS       YES
 1090   FCD4  EA                 NOP
 1091   FCD5  EA                 NOP
 1092   FCD6  EA                 NOP
 1093   FCD7  EA                 NOP
 1094   FCD8                ; FOR THE CASE OF THE HARRIS BIU WHICH IS USED TO COLLECT STATUS MESSAGES
 1095   FCD8                ; REPLACE THE FOUR NOPS ABOVE WITH THE FOLLOWING CODE.
 1096   FCD8                ;
 1097   FCD8                ; C9 00  CMP #STATUS     IS IT A STATUS PACKET? (*******)
 1098   FCD8                ; F0 08  BEQ FORUS       YES, SO GET IT
 1099   FCD8                ;
 1100   FCD8  A9 58      SKIPIT  LDA #%01011000  CAN'T RECEIVE, SO DISABLE RECEIVER
 1101   FCDA  8D 00 0C           STA MUARTS
 1102   FCDD  4C 9B FD           JMP NICLRX      AND RETURN
 1103   FCE0  24 02      FORUS   BIT INTBC       WAS THERE A PARITY ERROR IN THE ADDRESS?
 1104   FCE2  70 F4              BVS SKIPIT      YES, SO IGNORE THE PACKET
 1105   FCE4  A4 11              LDY INTSET      ARE WE ALL SET UP FOR INPUT?
 1106   FCE6  D0 10              BNE GETIT       YES
 1107   FCE8  A2 2D              LDX #NMLOST     INCR THE # OF MSGS LOST BECAUSE OF NO BUFFER
 1108   FCEA  20 F0 FD           JSR SETVC
 1109   FCED  A9 58              LDA #%01011000  TURN OFF THE RECEIVER
 1110   FCEF  8D 00 0C           STA MUARTS
 1111   FCF2                ;
 1112   FCF2                ;  SET UP TO SEND AN $FF ACK ...
 1113   FCF2                ;
 1114   FCF2  A9 FF              LDA #$FF
 1115   FCF4  85 3D              STA ACKBYT      TO SIGNAL THAT WE ARE FULL AND THAT HE SHOULD WAIT
 1116   FCF6  D0 52              BNE SNDACK      SEND THE STOP-XMIT FLAG (ALWAYS BRANCH)
 1117   FCF8                ;
 1118   FCF8  A0 00      GETIT   LDY #00         STORE BYTE 0 IN BUFFER
 1119   FCFA  91 00              STA (INPTR),Y
 1120   FCFC  85 3D              STA ACKBYT       SET UP TO SEND A GOOD ACK PACKET
 1121   FCFE  85 31              STA PARITY      START THE VERTICAL PARITY CALCULATION
 1122   FD00  C8         GETBYT  INY
 1123   FD01  AD 00 0C   WAITBY  LDA MUARTS      IS ANOTHER WORD READY
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  1124  FD04  29 05             AND #%00000101  IS RECEIVE KEY ON BUT WORD NOT IN?
  1125  FD06  F0 F9             BEQ WAITBY      IF NOT, GO BACK AND WAIT
  1126  FD08  29 04             AND #%00000100  IS RECEIVE KEY ON?
  1127  FD0A  D0 CC             BNE SKIPIT      IF NOT, TURN OFF THE RECEIVER
  1128  FD0C  2C 00 0C          BIT NUARTS      WAS THERE A PARITY ERROR?
  1129  FD0F  70 32             BVS PARERR      YES, SO ABORT THIS MESSAGE
  1130  FD11  AD 01 0C          LDA NUARTD      READ THE NEXT WORD
  1131  FD14  91 00             STA (INTPTR),Y
  1132  FD16  AA                TAX             SAVE THE DATA CHAR
  1133  FD17  45 31             EOR PARITY      ACCUMULATE THE PARITY
  1134  FD19  85 31             STA PARITY
  1135  FD1B  C0 07             CPY #07         IS THIS PACKET THE BYTE COUNT?
  1136  FD1D  90 E1             BCC GETBYT      IF <, KEEP ON READING IN THE HEADER
  1137  FD1F  D0 02             BNE INLOOP      IF >, COMPARE TO INTHC
  1138  FD21  86 02             STX INTHC       IF =, STORE RECEIVE PACKET LENGTH
  1139  FD23  C4 02      IN_LOOP CPY INTHC      DOES Y = PACKET LENGTH?
  1140  FD25  D0 D9             BNE GETBYT      NO, SO KEEP ON GETTING BYTES
  1141  FD27             ;
  1142  FD27  AD 00 0C   CHKLOP  LDA NUARTS      WAIT FOR THE CHECKSUM
  1143  FD2A  29 05             AND #%00000101
  1144  FD2C  F0 F9             BEQ CHKLOP
  1145  FD2E  29 04             AND #%00000100
  1146  FD30  D0 A6             BNE SKIPIT
  1147  FD32  2C 00 0C          BIT NUARTS      BAD PARITY?
  1148  FD35  70 0C             BVS PARERR      YES
  1149  FD37  AE 01 0C          LDX NUARTD      GET THE VERTICAL PARITY
  1150  FD3A  A9 58             LDA #%01011000  DISABLE THE RECEIVER
  1151  FD3C  8D 00 0C          STA NUARTS
  1152  FD3F  E4 31             CPX PARITY      COMPARE THE PARITIES
  1153  FD41  F0 07             BEQ SNDACK      HURRAH, THEY AGREE
  1154  FD43  A2 23      PARERR  LDX #$23CRC     KEEP COUNT OF PARITY ERRORS
  1155  FD45  20 F0 F0          JSR SERVC
  1156  FD48  D0 BE             BNE SKIPIT      DISABLE THE RECEIVER AND RETURN
  1157  FD4A             ;
  1158  FD4A  AD 00 0C   SNDACK  LDA NUARTS      IS THE NET BUSY?
  1159  FD4D  29 04             AND #%00000100
  1160  FD4F  F0 F9             BEQ SNDACK      YES
  1161  FD51  A9 18             LDA #%00011000  TURN ON THE XMITTER
  1162  FD53  8D 00 0C          STA NUARTS
  1163  FD56  A5 30             LDA ACKBYT      SEND THE ACK
  1164  FD58  8D 01 0C          STA NUARTD
  1165  FD5B  A2 1A             LDX #26         WAIT 75 US BEFORE TURNING OFF THE XMIT KEY
  1166  FD5D  CA         WAIT75  DEX
  1167  FD5E  10 FD             BPL WAIT75
  1168  FD60  A2 58             LDX #%01011000  TURN OFF THE XMITTER
  1169  FD62  8E 00 0C          STX NUARTS
  1170  FD65  C9 FF             CMP #$FF        DID WE SEND THE STOP-XMIT FLAG?
  1171  FD67  F0 2F             BEQ TRYACK      YES, GO TRY ALLOCATION & RETURN
  1172  FD69             ;
  1173  FD69             ; CHECK TO SEE IF THIS WAS A STATUS MESSAGE
  1174  FD69             ;
  1175  FD69  EA                 NOP
  1176  FD6A  EA                 NOP
  1177  FD6B             ; FOR THE CASE OF THE HARRIS BIU WHICH COLLECTS STATUS PACKETS REPLACE THE
  1178  FD6B             ; TWO NOPS ABOVE WITH THE FOLLOWING CODE.
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
 1179   FD6B              ; C9 00   CMP #STATUS     DID WE JUST PROCESS A STATUS MESSAGE?
 1180   FD6B  F0 25               BEQ STAT1       YES, SO DO WHAT HAS TO BE DONE
 1181   FD6D              ;
 1182   FD6D  A0 02               LDY #02         GET THE XMIT ADDR
 1183   FD6F  B1 00               LDA (INTPTR),Y
 1184   FD71  C5 34               CMP LASTTX      SAME AS THE PREVIOUSLY RECEIVED MESSAGE?
 1185   FD73  D0 08               BNE MSGOK       NO, SO THE MSG IS OK
 1186   FD75  A0 04               LDY #04         GET THE SEQUENCE #
 1187   FD77  B1 00               LDA (INTPTR),Y
 1188   FD79  C5 35               CMP LASTSQ      SAME AS THE PREVIOUS MSG?
 1189   FD7B  F0 1E               BEQ NICLRX      YES, SO SKIP IT
 1190   FD7D              ;
 1191   FD7D  A0 02       MSGOK   LDY #02         SAVE THE NEW TRANSMIT ADDRESS
 1192   FD7F  B1 00               LDA (INTPTR),Y
 1193   FD81  85 34               STA LASTTX
 1194   FD83  A0 04               LDY #04
 1195   FD85  B1 00               LDA (INTPTR),Y  THE NEW SEQUENCE #,
 1196   FD87  85 35               STA LASTSQ
 1197   FD89              ;
 1198   FD89  A9 16               LDA #QIN-NEXT   QUEUE THE MESSAGE TO BE SENT IN
 1199   FD8B  A0 0C               LDY CURNET
 1200   FD8D  20 14 FE            JSR FIQ
 1201   FD90  C6 11               DEC INTSET      NO LONGER SET UP FOR INPUT
 1202   FD92              ;
 1203   FD92              ; FOR NOW, ALL STATUS MESSAGES ARE IGNORED.
 1204   FD92              ;
 1205   FD92  EA          STAT1   NOP
 1206   FD93  A2 29               LDX #GMGCRC     INCR # OF MESSAGES RECEIVED WITH GOOD CKSUM
 1207   FD95  20 FD FD            JSR SFINC
 1208   FD98              ;
 1209   FD98  20 44 FC    TXNACK  JSR INTBUF      TRY TO GET SET UP AGAIN
 1210   FD9B  A2 00       NICLRX  LDX #00
 1211   FD9D  86 39       NIRET   STX INTX        SAVE OUR COPY OF X, OR CLEAR IT, AS NEEDED
 1212   FD9F  A2 00               LDX #00
 1213   FDA1  86 39               STX INTFLG
 1214   FDA3              ;
 1215   FDA3              ; THIS CODE IS USED TO RESTORE REGISTERS AND RETURN FROM INTERRUPTS.
 1216   FDA3              ;
 1217   FDA3  68          DORET   PLA             UNSTACK AND RETURN
 1218   FDA4  AA                  TAX
 1219   FDA5  68                  PLA
 1220   FDA6  A8                  TAY
 1221   FDA7  68                  PLA
 1222   FDA8  40                  RTI
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
 1224   FDA9              ;
 1225   FDA9              ; NMI OCCURS WHEN THE RECEIVE KEY TURNS OFF. (I. E. WHEN THE CARRIER ON THE
 1226   FDA9              ; CABLE GOES LOW.)
 1227   FDA9              ;
 1228   FDA9  48          NMI     PHA               PUSH A
 1229   FDAA  A9 58               LDA #%01011011    RESET NETWORK UART
 1230   FDAC  8D 00 0C            STA NUARTS
 1231   FDAF  A9 D8               LDA #%11011000    INITIALIZE NETWORK UART
 1232   FDB1  8D 00 0C            STA NUARTS
 1233   FDB4  A9 00               LDA #00           SET XMIT FLAG TO 0
 1234   FDB6  85 14               STA XMIT
 1235   FDB8  68                  PLA
 1236   FDB9  40                  RTI
```

```
CARD # LOC   CODE       CARD  10        20        30        40        50        60        70
 1238 FD8A             ;
 1239 FD8A             ; PCONST IS USED TO CONSTRUCT AN OUTGOING PACKET.
 1240 FD8A             ; IT EXPECTS THE MESSAGE TYPE TO BE PASSED IN A.
 1241 FD8A             ; IF IT CAN'T ALLOCATE A BUFFER, IT RETURNS WITH THE NEGATIVE BIT SET.
 1242 FD8A             ;
 1243 FD8A  78         PCONST  SEI             DON'T WANT INTERRUPTS DURING ALLOC
 1244 FD8B  20 0A FE           JSR ALLOC       GET A BUFFER
 1245 FD8E  58                 CLI
 1246 FD8F  30 29              BMI PRET        COULDN'T GET A BUFFER, SO RETURN
 1247 FDC1  A4 71              LDY LOPTR,X     GET THE PTR TO THE BUFFER
 1248 FDC3  84 0A              STY BIUPTR
 1249 FDC5  A4 87              LDY HIPTR,X
 1250 FDC7  84 0B              STY BIUPTR+1
 1251 FDC9  86 0E              STX CURBIU      SAVE THE BUFFER #
 1252 FDC9  AA                 TAX
 1253 FDCC  A9 00              LDA #00         CLEAR THE MESSAGE HEADER
 1254 FDCE  A0 06              LDY #06
 1255 FDD0  91 0A      CLOOP   STA (BIUPTR),Y
 1256 FDD2  88                 DEY
 1257 FDD3  D0 FB              BNE CLOOP
 1258 FDD5             ;
 1259 FDD5  A5 3E              LDA XADDR
 1260 FDD7  91 0A              STA (BIUPTR),Y  STORE THE DESTINATION ADDRESS (Y = 0)
 1261 FDD9  A0 02              LDY #02
 1262 FDDB  AD 40 10           LDA PORT3B      FOR THE NEW BIU'S THE HOME ADDRESS IS SET HERE
 1263 FDDE  91 0A              STA (BIUPTR),Y  STORE THE TRANSMIT ADDRESS
 1264 FDE0  8A                 TAA
 1265 FDE1  A0 05              LDY #05
 1266 FDE3  91 0A              STA (BIUPTR),Y  STORE THE MESSAGE TYPE
 1267 FDE5  A9 07              LDA #07         DEFAULT PACKET LENGTH OF 7
 1268 FDE7  AA                 TAY
 1269 FDE8  91 0A              STA (BIUPTR),Y  STORE IT IN THE PACKET
 1270 FDEA  60         PRET    RTS
 1271 FDEB             ;
 1272 FDEB             ; STIMER SETS A TIMER VARIABLE TO AN INCREMENT PLUS THE CURRENT
 1273 FDEB             ; VALUE OF TOD. A IS THE INCREMENT TO ADD TO TOD AND X POINTS TO
 1274 FDEB             ; THE VARIABLE THAT HOLDS THE RESULT, LOW BYTE FIRST. A IS CHANGED.
 1275 FDEB             ;
 1276 FDEB  18         STIMER  CLC
 1277 FDEC  65 18              ADC TOD         ADD TOD TO THE INCR IN A
 1278 FDEE  95 00              STA 0,X         SAVE IT IN THE LOW RESULT BYTE
 1279 FDF0  A5 19              LDA TOD+1       ADD IN THE CARRY
 1280 FDF2  69 00              ADC #00
 1281 FDF4  95 01              STA 1,X         SAVE IT IN THE SECOND BYTE
 1282 FDF6  A5 1A              LDA TOD+2
 1283 FDF8  69 00              ADC #00
 1284 FDFA  95 02              STA 2,X         AND IN THE HIGH BYTE
 1285 FDFC  60                 RTS
```

```
CARD # LOC  CODE      CARD 10        20        30        40        50        60        70
 1287  FDFD                ;
 1288  FDFD                ; SFINC IS CALLED TO INCREMENT THE CONTENTS OF A STATUS MSG FIELD. THE
 1289  FDFD                ; FIELD WILL NOT WRAP AROUND TO 0000. X => THE FIELD (HI BYTE FIRST).
 1290  FDFD                ;
 1291  FDFD  F6 01         SFINC   INC 1,X         INCREMENT THE LOWER BYTE
 1292  FDFF  D0 08                 BNE SFRET       IF NO CARRY, ALL DONE
 1293  FE01  F6 00                 INC 0,X         INCREMENT THE HIGHER BYTE
 1294  FE03  D0 04                 BNE SFRET       IF NOT ZERO, THEN ALL DONE
 1295  FE05  D6 01                 DEC 1,X         WRAPPED AROUND, SO RETURN TO FFFF
 1296  FE07  D6 00                 DEC 0,X
 1297  FE09  60            SFRET   RTS
 1298  FE0A                ;
 1299  FE0A                ; ALLOC IS CALLED TO ALLOCATE A FREE BUFFER AND RETURN ITS NUMBER IN
 1300  FE0A                ; X. IF THERE ARE NONE LEFT, THE N BIT WILL BE ON. X AND Y ARE CHANGED.
 1301  FE0A                ;
 1302  FE0A  A4 36         ALLOC   LDY STKPTR      GET OFFSET OF THE TOP OF FREE BUFFER STACK + 1
 1303  FE0C  88                    DEY             POINT TO THE NEXT FREE BUFFER NUMBER
 1304  FE0D  30 04                 BMI ALRET       RETURN IF NONE AVAILABLE
 1305  FE0F  94 36                 STY STKPTR      GOT ONE, SO SAVE THE NEW TOP OF THE STACK
 1306  FE11  B6 53                 LDX BUFSTK,Y    GET THE ALLOCATED BUFFER NUMBER
 1307  FE13  60            ALRET   RTS
 1308  FE14                ;
 1309  FE14                ; ENQ IS USED TO ADD A BUFFER TO THE END OF A QUEUE.  A IS THE OFFSET OF
 1310  FE14                ; THE QUEUE HEADER FROM NEXT, Y IS THE # OF THE BUFFER.
 1311  FE14                ;
 1312  FE14  08            ENQ     PHP             TURN OFF INTERRUPTS
 1313  FE15  78                    SEI
 1314  FE16  AA            ENQ1    TAX             STEP ONE FURTHER ALONG THE QUEUE
 1315  FE17  B5 42                 LDA NEXT,X      IS THIS THE LAST ENTRY?
 1316  FE19  10 F8                 BPL ENQ1        NO, SO KEEP ON LOOKING
 1317  FE1B  94 42                 STY NEXT,X      SET LINK IN LAST ENTRY TO NEW ENTRY
 1318  FE1D  99 42 00              STA NEXT,Y      SET THE NEXT PTR IN THE NEW ENTRY TO NULL
 1319  FE20  28                    PLP
 1320  FE21  60                    RTS
 1321  FE22                ;
 1322  FE22                ; DQ IS CALLED TO DEQUEUE AND FREE A BUFFER ON A QUEUE. X IS THE OFFSET
 1323  FE22                ; FROM NEXT OF ENTRY PREVIOUS TO THE ONE TO BE FREED. X,A,Y ARE CHANGED
 1324  FE22                ; THIS DQ DIFFERS FROM THE TMS 910 DQ IN THAT IT LACKS THE FREE
 1325  FE22                ; ENTRY POINT
 1326  FE22                ;
 1327  FE22  08            DQ      PHP
 1328  FE23  78                    SEI             TURN OFF INTERRUPTS.
 1329  FE24  B4 42                 LDY NEXT,X      Y HAS THE NUMBER OF THE BUFFER TO BE FREED
 1330  FE26  B9 42 00              LDA NEXT,Y      A HAS THE NUMBER OF THE NEXT BUFFER IN THE QUEUE
 1331  FE29  95 42                 STA NEXT,X      THE BUFFER IS DEQUEUED
 1332  FE2B  A6 36                 LDX STKPTR      ADD THE BUFFER TO THE FREE BUFFER STACK
 1333  FE2D  98                    TYA
 1334  FE2E  95 53                 STA BUFSTK,X    NOW IN THE FREE BUFFER POOL
 1335  FE30  E6 36                 INC STKPTR      ONE MORE FREE BUFFER
 1336  FE32  28                    PLP
 1337  FE33  60                    RTS
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
 1339   FE34              ;
 1340   FE34              ; CONSTANTS FOLLOW
 1341   FE34              ;
 1342   FE34              ; THE FOLLOWING 12-BYTE BIJ FUNCTION DESCRIPTION IS PLACED INTO EACH
 1343   FE34              ; BIU STATUS MESSAGE.
 1344   FE34              ;
 1345   FE34  48 41       BIJFNC   .BYTE  'HARRIS BIJ  '
 1345   FE36  52 52
 1345   FE38  49 53
 1345   FE3A  20 42
 1345   FE3C  49 55
 1345   FE3E  20 20
 1346   FE40              ;
 1347   FE40                       *=$FFFA
 1348   FFFA  A9 FD       VECTOR   .WORD NMI        NON-MASKABLE INTERRUPT VECTOR
 1349   FFFC  00 F8                .WORD RESET      RESET VECTOR
 1350   FFFE  5D FC                .WORD IRQ        IRQ VECTOR
 1351   0000              ;
 1352   0000                        .END
```

END OF [illegible]TECHNOLOGY 650X ASSEMBLY VERSION 5.1
NUMBER OF ERRORS = 0, NUMBER OF WARNINGS = 0

SYMBOL TABLE

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
ABUF	FA06	548	543
ACKBYT	003D	134	1115 1120 1163
ACKOK	F9E0	510	504
ACKCK	F9D1	502	481
AKLOOP	F992	480	493
ALLOC	FE0A	1302	1003 1244
ALRET	FE15	1307	1304
ANDONE	F9F1	521	522
BADACK	F98A	487	506 508
BIUFNC	FE34	1345	968
BIUPTR	000A	99	548 550 909 912 917 934 948 952 955 958 961 969 1248 1250 1255 1260 1263 1266 1269
BJFCNT	0016	87	148 158 162 163 232 242
BJFLEN	0040	85	87 254
BJFSTK	005B	155	236 1306 1334
BJFFRI	F97E	249	258
BJFMEM	0800	86	87
BYTLOW	003B	132	609 645
BYTRET	FA7C	646	663 683
BYTES	FA76	642	635
COM	0010	63	563
CHARS	FA65	630	636
CHKLOP	FD27	1142	1144
CKCRET	F9AA	867	865 906
CKSTAT	F89E	863	851 854
CKTOUT	F9BA	850	352
CLOOP	FD00	1255	1257
COLIDE	FCAE	1069	1062
CONECT	0040	142	276 304 505 537 692 765 775 784 789 817 859
CONBB	F9E6	328	326
CONSOF	F8AF	284	282
CONTYS	F853	803	803
CTIME	F849	875	853 864
CTRET	F88D	885	877 878 881 882
CURNET	000C	100	1009 1199
CURDEV	0000	101	553 667
CURBIU	000E	102	285 329 552 814 982 1251
DATMSG	0000	17	541 690
DATU	0002	70	745
DCRD	FC7A	1039	1036
DCRET	FDA3	1217	1040
DISC	0032	77	1028
DMAMOD	FA5A	595	589
DMATM	005C	133	581 596
DMATTY	005A	154	227 575 588 594 650 659
DMAIT	FA36	593	653
DQ	FE22	1327	517 822
DUARTS	1400	28	954
ENQ1	FE16	1314	1316
ENQ	FE14	1312	286 330 668 815 983 1200
FFLOOP	F961	229	230
FORUS	FCE0	1103	1089
GETBYT	FD00	1122	1136 1140

```
SYMBOL:  VALUE  LINE DEFINED        CROSS-REFERENCES

GETDAT   FA32        589   568
GETCH    FA18        562   540   582   625   637
GETLOP   FA58        619   620
GETCDA   FA7D        650   565
GETIT    FCF8       1118  1106
HARADM   000F         11   337   611  1035
HARGO    003F        141   273   292   298
HARADR   002F         12   733
HIPTR    0087        163   243   251   401   686  1007  1249
HOLDTX   F94C        464   465
IIFJ     0008         78   610   732
INBC     0005         95   556   562   630   634   642   773
INDONE   FAB4        658   586   651
INDON1   FAB9        661   633
INDEV    F9F6        537   350   670
INLOOP   FD23       1139  1137
INPTR    0003         94   549   551   631   665   804
INSET    000F        103   539   554   669   802
INTX     0038        129  1050  1211
INTFLG   0039        130   391   415  1213
INTBC    0002         93  1049  1074  1077  1103  1138  1139
INTBUF   FC44        999   339   349  1209
INTPTR   0000         92  1006  1008  1119  1131  1183  1187  1192  1195
INTSET   0011        105  1001  1010  1105  1201
INTRES   FC5B       1011  1002  1004
IRET     FA05        544   538   576
IRQ      FC5D       1020  1350
KEYVOF   F9A4        469   470
LASTTX   0034        125  1184  1193
LASTSQ   0035        126  1188  1196
LOPTR    0071        162   245   249   252   399   684  1005  1247
LWAIT    F916        375   384
MAXPAX   0017        110   323   632   809
MLOOP    F8F0        347   353
MSGOK    FD7D       1191  1185
NETBSY   F95E        419   394
NET      F912        361   347
NETFRE   F9E6        515   497
NETRET   F9F5        523   494
NEXT     0042        143   284   328   516   666   813   821   929   942   980  1198
                                1315  1317  1318  1329  1330  1331
NICLRX   FD9B       1210  1067  1086  1102  1189
NIIFU    00F8         79   621   749
NINT     FCB2       1047  1030
NIRET    FD9D       1211  1057
NMBCRC   0028        119  1154
NMCOL    0025        116   438  1084
NMDISC   0027        117   495
NMGCRC   0029        118  1206
NMI      FDA9       1228  1348
NMLOST   0020        120  1107
NMRXMT   0023        115   489
NMWAIT   002F        121   419
NMXMT    0021        114   511   916   990
NOSOF    F8D2        317   293   299   310
NOTPWR   F8CB        304   275
NOXMIT   0037        128   375   378   507
```

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
NRET	F963	421	362
NTPRX	0032	123	406 491 492 519
NTXCOL	F976	444	434
NJARTS	0C00	26	184 186 392 416 427 444 458 467 480 951 1047 1066 1070 1101 1110 1123 1128 1142 1147 1151 1158 1162 1169 1230 1232
NJARTD	0C01	27	448 462 502 1048 1130 1149 1164
NJLL	F808	757	723
NJLRET	FAE1	726	758
ODH	0002	62	566 624
OUTPTR	0006	96	400 402 405 408 410 447 476 685 687 689 702 719 725 769 793 1061
OUTDEV	FA9E	680	348
OUTBC	0008	97	411 451 477 503 701 705 748
OUTPL	0009	99	703 706 722
OUTLOP	FAF5	743	747
OUTFRE	F869	821	695 709 766 770 776 781 785 788 791 812
OUTSET	0010	104	680 699 708
P1ACR	101B	35	198
P1ADDR	1013	32	190
P1BDDR	1012	31	191
P1IER	101E	38	202
P1IFR	101D	37	564 567 830 957
P1PCR	101C	36	196
P2ACR	102B	43	193
P2ADDR	1023	42	209
P2BDDR	1022	41	210
P2IFR	102D	45	746
P2IER	102E	46	203
P2PCR	102C	44	198
P3ACR	104B	51	194
P3ADDR	1043	50	211 1032 1039
P3BDDR	1042	49	192
P3IER	104E	54	204 207
P3IFR	104D	53	1029
P3PCR	104C	52	200 617 622 735 750
PARERR	FD43	1154	1129 1148
PARITY	0031	122	389 449 450 461 1121 1133 1134 1152
PCONST	FD8A	1243	281 325 542 811 905
PDATA	FA4F	699	694
PORT1B	1010	29	599
PORT2B	1020	39	727
PORT2A	1021	40	728
PORT3B	1040	47	260 1088 1262
PORT3A	1041	48	338 612 734 1034 1037
PORT1A	1011	30	607
PREPAR	F92A	389	381
PRET	FDEA	1270	1246
PUTCH	FAC9	705	681 752
QIN	0058	152	227 228 682 821 937 1198
QOUT	0059	153	284 328 361 516 656 813 924 980
RAMSIZ	0C00	84	86
RANDU	0012	106	261 1071 1079 1082
RESET	F800	181	283 327 1349
RNDCNT	0013	107	379 380 382 1083
RXDATA	FCCF	1089	1032
SACMSG	000F	20	778 810

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
SBOT2	F8F1	934	925
SBOT3	FC05	947	938
SFINC	FDF0	1291	420 439 490 496 512 1085 1108 1155 1207
SFRET	FE09	1297	1292 1294
SKIPI	F88B	257	255
SKIPIT	FCDB	1100	1104 1127 1146 1156
SLOOP1	F8D5	916	921
SLOOP2	F8E5	926	933
SLOOP3	F8F9	939	946
SLOOP5	FC3E	990	992
SLOOP4	FC21	968	973
SNDSTA	F8AE	890	866
SNDSOF	F8A5	280	297 311
SNDACK	FD4A	1158	1116 1153 1160
SOFMSG	00DE	19	280 763
SONMSG	00E0	21	324 787
SONUM	0033	124	403 520
STAT1	F092	1205	1190
STACK1	F86B	255	238
STATUS	0000	10	908
STAMSG	00DB	18	890
STIMER	FDEB	1276	321 986
STKPTR	0036	127	233 1302 1305 1332 1335
SWAIT2	F924	382	383
SWAIT1	F920	380	376
TESTTX	FC9A	1059	1056
TEST2	FC9C	1060	1064
TICK	0041	144	216 833 836
TIMRH	1015	34	214
TIMRL	1014	33	212 832
TIMOUT	F86F	830	351
TIMRET	F889	842	831 834 838 840
TOD	0018	111	837 839 841 875 879 883 1277 1279 1282
TOKEN	0015	109	1053 1060
TRMCNT	0008	14	296
TRMBAS	0051	13	277 296
TRNSMT	F964	426	418
TRYSAC	F82A	778	764
TRYSOF	F80F	765	691
TRYDS	FAD4	719	707
TRYLAT	F801	749	744
TRYSON	F839	787	779
TSACK	001E	113	320 852
TSTAT	001B	112	263 863 985
TTY	0002	22	541 690
TXBEMP	F96C	433	446
TXCOL	F970	438	457
TXLOOP	F96B	432	452
TXNACK	FD9B	1209	1171
TXPAR	F98A	456	460
VECTOR	FFFA	1348	****
WAITBY	FD01	1123	1125
WAIT75	FD5D	1165	1167
WAIT	003A	151	331 780 783 796 850 856
WAT100	F952	413	414
XADDR	003E	140	218 278 294 295 318 767 772 794 857 960 1259

```
SYMBOLI  VALUE  LINE DEFINED        CROSS-REFERENCES

XMIT     0014        108   428  433  456  469  478  487  521 1051 1234
ZLOOP    F85A        221   223
```

APPENDIX II

GRAPHIC 7 BIU SOURCE LISTING

This is the source listing for the Graphic 7 BIU software. The reader is directed to Sections VII and IX for a detailed description of the operation of this code.

```
CARD #  LOC   CODE   CARD  10        20        30        40        50        60        70
     2  0000           .OPT NOCNT,XREF,MEM,LIST,ERR,GEN
     3  0000         ;
     4  0000         ; THIS IS THE GRAPHIC 7 BIJ CODE AS OF 10/31/79 WITH FLOW CONTROL.
     5  0000         ;
     6  0000         HARBAS   = $21          BASE ADDRESS FOR HARRIS BIUS
     7  0000         TRMCNT   = $08          NUMBER OF G7 TERMINALS IN CAPS SYSTEM
     8  0000         HARADR   = $1F          UNIT CODE REGISTER ADDRESS AND DATA DIRECTION BIT
     9  0000         PRIMER   = $19          PRIME THE CONNECT/DISCONNECT LINES
    10  0000         ; MESSAGE TYPE EQUATES
    11  0000         ;
    12  0000         DATMSG   = 0            DATA MESSAGES ARE 00-1F
    13  0000         STAMSG   = $D8          STATUS MESSAGE
    14  0000         SOFMSG   = $DE          SIGN-OFF MESSAGE
    15  0000         SACMSG   = $DF          SIGN-ON ACKNOWLEDGMENT
    16  0000         SONMSG   = $E0          SIGN-ON MSG (E0-FF)
    17  0000         TTY      = 2            DATA TYPE FOR TTY-COMPATIBLE TERMINALS
    18  0000         ;
    19  0000         ; DEVICE ADDRESSES FOLLOW
    20  0000         ;
    21  0000         NUARTS   = $C00         NETWORK UART STATUS
    22  0000         NUARTD   = $C01         NETWORK UART DATA
    23  0000         DUARTS   = $1000        DEVICE UART STATUS
    24  0000         PORT1B   = $1010        PARALLEL PORT 1B DATA REGISTER
    25  0000         PORT1A   = $1011        PARALLEL PORT 1A DATA REGISTER
    26  0000         P1BDDR   = $1012        PARALLEL PORT 1B DATA DIRECTION REGISTER
    27  0000         P1ADDR   = $1013        PARALLEL PORT 1A DATA DIRECTION REGISTER
    28  0000         T1MRL    = $1014        TIMER LOW BYTE
    29  0000         T1MRH    = $1015        TIMER HIGH BYTE
    30  0000         P1ACR    = $101B        PARALLEL PORT 1 AUXILIARY CONTROL REGISTER
    31  0000         P1PCR    = $101C        PARALLEL PORT 1 PERIPHERAL CONTROL REGISTER
    32  0000         P1IFR    = $101D        PARALLEL PORT 1 INTERRUPT FLAG REGISTER
    33  0000         P1IER    = $101E        PARALLEL PORT 1 INTERRUPT ENABLE REGISTER
    34  0000         PORT2B   = $1020        PARALLEL PORT 2B DATA REGISTER
    35  0000         PORT2A   = $1021        PARALLEL PORT 2A DATA REGISTER
    36  0000         P2BDDR   = $1022        PARALLEL PORT 2B DATA DIRECTION REGISTER
    37  0000         P2ADDR   = $1023        PARALLEL PORT 2A DATA DIRECTION REGISTER
    38  0000         P2ACR    = $102B        PARALLEL PORT 2 AUXILIARY CONTROL REGISTER
    39  0000         P2PCR    = $102C        PARALLEL PORT 2 PERIPHERAL CONTROL REGISTER
    40  0000         P2IFR    = $102D        PARALLEL PORT 2 INTERRUPT FLAG REGISTER
    41  0000         P2IER    = $102E        PARALLEL PORT 2 INTERRUPT ENABLE REGISTER
    42  0000         PORT3B   = $1040        PARALLEL PORT 3B DATA REGISTER
    43  0000         PORT3A   = $1041        PARALLEL PORT 3A DATA REGISTER
    44  0000         P3BDDR   = $1042        PARALLEL PORT 3B DATA DIRECTION REGISTER
    45  0000         P3ADDR   = $1043        PARALLEL PORT 3A DATA DIRECTION REGISTER
    46  0000         P3ACR    = $1048        PARALLEL PORT 3 AUXILIARY CONTROL REGISTER
    47  0000         P3PCR    = $104C        PARALLEL PORT 3 PERIPHERAL CONTROL REGISTER
    48  0000         P3IFR    = $104D        PARALLEL PORT 3 INTERRUPT FLAG REGISTER
    49  0000         P3IER    = $104E        PARALLEL PORT 3 INTERRUPT ENABLE REGISTER
```



```
CARD #  LOC   CODE      CARD  10        20        30        40        50        60        70
   51   0000           ;
   52   0000           ; EQUATES FOR PORT 1 HANDSHAKING FLAGS IN P1IFR
   53   0000           ;
   54   0000           ;         *** ICTL IS THE SIGNAL ON THE CA2 LINE OF THE 6522/1  ***
   55   0000           ;         *** TO INDICATE THE BIU HAS TAKEN G7 DATA (AUTOMATIC) ***
   56   0000           ;
   57   0000           IWR      = $02           1 => GRAPHIC 7 HAS DATA READY (DAVFJ ***CA1***)
   58   0000           ;
   59   0000           ; EQUATES FOR PORT 2 HANDSHAKING FLAGS IN P2IFR
   60   0000           ;
   61   0000           ;         *** OCTL IS THE SIGNAL ON CA2 OF 6522/2 TO INDICATE   ***
   62   0000           ;         ***   THAT THE BIU HAS DATA FOR THE G7 (AUTOMATIC)    ***
   63   0000           ;
   64   0000           OWR      = $02           1 => GRAPHIC 7 HAS ACCEPTED DATA FROM BIU (DDACP)
   65   0000           ;
   66   0000           ; EQUATES FOR PORT 3 INPUT/OUTPUT SIGNAL LINES
   67   0000           ;
   68   0000           ;         *** DISC IS A PULSE SENT BY THE BIU ON CA1 OF 6522/3  ***
   69   0000           ;         *** WHICH SIGNALS THE G7 TO DROP CNCT AND READ PORT3A ***
   70   0000           ;
   71   0000           CNCT     = $02           GRAPHIC 7 SIGNAL TO BIU IT IS CONNECTED
   72   0000           ;
   73   0000           ; THE NEXT EQUATES REFLECT THE ACTUAL OPERATIONAL PARAMETERS
   74   0000           ; OF THE BIU WITH RESPECT TO BUFFER COUNT AND SIZE
   75   0000           ;
   76   0000           RAMSIZ   = 3072          NUMBER OF BYTES OF RAM AVAILABLE
   77   0000           BUFLEN   = 128           NUMBER OF BYTES IN A BUFFER (DO NOT CHANGE)
   78   0000           BUFMEM   = RAMSIZ-256    AMOUNT OF MEMORY ALLOCATED TO BUFFERS
   79   0000           BUFCNT   = BUFMEM/BUFLEN NUMBER OF BUFFERS AVAILABLE
```

```
CARD #  LOC   CODE     CARD  10        20        30        40        50        60        70
    81  0000           ;
    82  0000           ; THE FOLLOWING VARIABLES ARE CLEARED WHENEVER RESET IS HIT.
    83  0000           ;    (DOWN TO BUT NOT INCLUDING 'XADDR')
    84  0000           ;
    85  0000  00 00    INTPTR   .DBYTE 0        BUFFER PTR USED BY IRQ
    86  0002  00       INTBC    .BYTE  0        BYTE COUNT USED BY IRQ
    87  0003  00 00    INPTR    .DBYTE 0        BUFFER PTR USED BY INDEV
    88  0005  00       INBC     .BYTE  0        BYTE COUNT USED BY INDEV
    89  0006  00 00    OUTPTR   .DBYTE 0        BUFFER PTR USED BY OUTDEV AND NET
    90  0008  00       OUTBC    .BYTE  0        BYTE COUNT USED BY OUTDEV AND NET
    91  0009  00       OUTPL    .BYTE  0        LENGTH OF PACKET BEING SENT TO DEVICE
    92  000A  00 00    BIUPTR   .DBYTE 0        BIU-CREATED BUFFER POINTER
    93  000C  00       CURNET   .BYTE  0        THE CURRENT PACKET BEING READ IN FROM THE NET
    94  000D  00       CURDEV   .BYTE  0        THE CURRENT PACKET COMING FROM THE DEVICE
    95  000E  00       CURBIU   .BYTE  0        THE CURRENT PACKET COMING FROM THE BIU
    96  000F  00       INSET    .BYTE  0        WHETHER A DEVICE INPUT BUFFER IS SET UP
    97  0010  00       OUTSET   .BYTE  0        WHETHER A DEVICE OUTPUT BUFFER IS SET UP
    98  0011  00       INTSET   .BYTE  0        WHETHER A NETWORK INPUT BUFFER IS SET UP
    99  0012  00       RANDU    .BYTE  0        CURRENT RANDOM NUMBER
   100  0013  00       RNDCNT   .BYTE  0        CURRENT RANDOM COUNT
   101  0014  00       XMIT     .BYTE  0        TRANSMIT FLAG
   102  0015  00 00    TOKEN    .DBYTE 0        TEMPORARY RECEIVED DATA BUFFER
   103  0017  00       WAIT     .BYTE  0        WAITING FOR SIGN ON ACK FLAG
   104  0018  00       MAXPAX   .BYTE  0        THE CURRENT MAX # OF CHARS IN A BUFFER
   105  0019           TOD      *=*+3           THE CURRENT TIME OF DAY IN SECONDS.
   106  001C           TSTAT    *=*+3           TIME TO XMIT THE NEXT STATUS MSG
   107  001F           TSACK    *=*+3           TIMEOUT ON WAITING FOR SIGN-ON ACK
   108  0022           TYACK    *=*+3           TIMEOUT ON ANSWER TO TERMINAL SIGN-ON REQ
   109  0025           TETTY    *=*+3           TTY TIMER CHECK
   110  0028  00 00    NMXMT    .DBYTE 0        # OF PACKETS TRANSMITTED SUCESSFULLY
   111  002A  00 00    NMRXMT   .DBYTE 0        # OF PACKETS RE-TRANSMITTED BECAUSE NO ACK
   112  002C  00 00    NMCOL    .DBYTE 0        # OF COLLISIONS AND RETRANSMISSIONS
   113  002E  00 00    NMDISC   .DBYTE 0        # OF PACKETS DISCARDED
   114  0030  00 00    NMGCRC   .DBYTE 0        # OF PACKETS RECEIVED WITH GOOD CRC
   115  0032  00 00    NMBCRC   .DBYTE 0        # OF PACKETS RECEIVED WITH BAD CRC
   116  0034  00 00    NMLOST   .DBYTE 0        # OF PACKETS LOST BECAUSE OF NO BUFFER SPACE
   117  0036  00 00    NMWAIT   .DBYTE 0        # OF TIMES BIU HAD TO WAIT TO XMIT
   118  0038  00       PARITY   .BYTE  0        USED TO FLAG BAD PARITY IN AN INCOMING PACKET
   119  0039  00       NTPRX    .BYTE  0        # OF TIMES THE CURRENT PACKET IS RETRANSMITTED
   120  003A  00       SJNJM    .BYTE  0        THE SEQUENCE # OF THE NEXT PACKET TO BE XMITTED
   121  003B  00       LASTTX   .BYTE  0        THE XMIT ADDR OF THE LAST PACKET RECEIVED
   122  003C  00       LASTSQ   .BYTE  0        THE SEQUENCE # OF THE LAST PACKET RECEIVED
   123  003D  00       STKPTR   .BYTE  0        INDEX OF TOP OF FREE BUFFER STACK
   124  003E  00       NOXMIT   .BYTE  0        # OF MILLISECONDS TO STOP XMITTING
   125  003F  00       INTX     .BYTE. 0        VALUE OF REGISTER X WHEN NINT IS NEXT ENTERED
   126  0040  00       INTFLG   .BYTE. 0        BYTE CLEARED BY NINT EVERY TIME IT IS ACTIVE
   127  0041  00       BYTLOW   .BYTE  0        USED TO STORE LOW ORDER BYTE OF WORD TEMPORARILY
   128  0042  00       DMATM    .BYTE  0        TIMER TO HOLD DMA FLAG ON DURING INPUT OPERATION
   129  0043  00       TTYTM    .BYTE  0        TTY TIMER FLAG
   130  0044  00       SUNCNT   .BYTE  0        COUNTER TO RECORD NUMBER OF HARRIS BIU SIGN-ONS
   131  0045  00       TERMF    .BYTE  0        FLAG USED TO INDICATE CONNECTED TO SERIAL TERM
   132  0046  00       ACKBYT   .BYTE  0        BYTE INDICATING THE VALUE TO BE USED IN ACK PACKET
   133  0047           STRING   *=*+5           INPUT STORAGE FOR 'WHICH SYSTEM?' RESPONSE
```

```
CARD #  LOC   CODE    CARD  10        20        30        40        50        60        70
  135   004C          ;
  136   004C          ; THE FOLLOWING VARIABLES ARE NOT CLEARED AT RESET TIME
  137   004C          ; (ALL VARIABLES PRECEDNG "XADDR" WILL BE SET TO ZERO.)
  138   004C          ;
  139   004C  00       XADDR    .BYTE  0        THE CURRENT XMIT ADDRESS
  140   004D  00       HARDO    .BYTE  0        INITIAL POWER UP FLAG
  141   004E  00       CONECT   .BYTE  0        1 => CONNECTED, 0 => WAITING FOR REPLY TO ...
  142   004F          ;     LINK REQUEST MESSAGE, -1 => WAITING FOR REPLY TO WHICH SYSTEM? MSG
  143   004F  00       TICK     .BYTE  0        COUNTS THE # OF CLOCK TICKS IN 1/4 SECOND (25)
  144   0050          ;
  145   0050          ; THE BUFFERS EACH HAVE AN ENTRY IN THIS ARRAY
  146   0050          ;
  147   0050          NEXT     *=*+BUFCNT      THE NEXT POINTER FOR EACH BUFFER
  148   0066          ;
  149   0066          ; THE QUEUE HEADERS FOLLOW
  150   0066          ;
  151   0066  00       QIN      .BYTE  0        THE QUEUE OF BUFFERS WAITING FOR GRAPHIC 7
  152   0067  00       QOUT     .BYTE  0        THE QUEUE OF BUFFERS WAITING FOR THE NETWORK
  153   0068          ;
  154   0068          ; CONSTANTS INITIALIZED TO $FF
  155   0068          ;
  156   0068  00       DMATTY   .BYTE  0        FLAG TO INDICATE WHICH COM MODE BIU IS IN...
  157   0069          ;                           ...(1=>DMA, 0=>TTY, -1=>NEW/UNDETERMINED)
  158   0069          ;
  159   0069          ; THE BUFFER ALLOCATION STACK, BELOW, CONTAINS THE FREE BUFFER LIST
  160   0069          ;
  161   0069          BUFSTK   *=*+BUFCNT      THE FREE BUFFER STACK
  162   007F          ;
  163   007F          ; THESE ARE THE POINTERS TO THE BUFFERS
  164   007F          ;
  165   007F          LOPTR    *=*+BUFCNT      THE LOW HALF OF THE PTRS
  166   0095          HIPTR    *=*+BUFCNT      THE HIGH HALF OF THE PTRS
  167   00AB          ;
  168   00AB          ; THE BUFFERS ARE LOCATED IN CONTIGUOUS MEMORY STARTING WITH BUFFER NUMBER 2
  169   00AB          ; AT ADDRESS 0180.  BUFFER NUMBER 1 IS LOCATED IN A GAP BETWEEN THE ZERO PAGE
  170   00AB          ; VARIABLES AND THE PROCESSOR STACK (00C0-013F).  THE APPROPRIATE ADDRESSES
  171   00AB          ; ARE LOADED INTO HIPTR/LOPTR BY THE RESET ROUTINE.
  172   00AB          ;
  173   00AB          ; THE PROCESSOR STACK IS SET TO START AT MEMORY LOCATION 017F AND WILL
  174   00AB          ; DECREMENT WHEN NECESSARY.  IT WILL USE RAM AREA BETWEEN 017F-0140 DURING
  175   00AB          ; JSR'S, PHP'S, PHA'S AND INTERRUPTS WHENEVER NECESSARY.  THIS SHOULD BE
  176   00AB          ; MORE THAN ENOUGH SPACE FOR THIS PURPOSE.
  177   00AB          ;
  178   00AB          ; (SEE CODE IN RESET ROUTINE BETWEEN STATEMENTS ZLOOP AND STACKI)
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  180   00AB
  181   00AB              ;
  182   00AB              ; THE CODE FOLLOWS. EXECUTION COMES HERE WHEN RESET IS HIT.
  183   00AB              ;
  184   00AB                        *=$F800
  185   F800  D8    *      RESET    CLD               DON'T WANT DECIMAL MODE
  186   F801  78                    SEI                 DON'T WANT INTERRUPTS IF WE GOT HERE FROM MAIN LOOP
  187   F802              ;
  188   F802  A2 5B                 LDX #%01011011    RESET NETWORK UART
  189   F804  8E 00 0C              STX NUARTS
  190   F807  A2 D9                 LDX #%11011000    INITIALIZE NETWORK UART
  191   F809  8E 00 0C              STX NUARTS
  192   F80C  A2 40                 LDX #%01000000    TIMER1 IS FREE RUNNING
  193   F80E  8E 14 10              STX P1ACR
  194   F811  A9 00                 LDA #%00000000    PORTS 1 & 3B LINES ARE ALL INPUTS
  195   F813  8D 13 10              STA P1ADDR
  196   F816  8D 12 10              STA P1BDDR
  197   F819  8D 42 10              STA P3BDDR
  198   F81C  8D 24 10              STA P2ACR         DISABLE PORTS 2 & 3 TIMERS AND LATCHES
  199   F81F  8D 44 10              STA P3ACR
  200   F822  A2 9B                 LDX #%10011011    PULSE HANDSHAKING (ACTIVE HIGH) FOR PORT 1
  201   F824  8E 1C 10              STX P1PCR
  202   F827  A2 88                 LDX #%10001000    LEVEL HANDSHAKE (ACTIVE LOW) ON PORT 2
  203   F829  8E 2C 10              STX P2PCR
  204   F82C  A2 EA                 LDX #%11101010    PULSE HANDSHAKING ON PORT 3A / MANUAL PORT 3B
  205   F82E  8E 4C 10              STX P3PCR
  206   F831  A2 7F                 LDX #%01111111    DISABLE ALL INTERRUPTS FROM PORTS 1,2 & 3
  207   F833  8E 1E 10              STX P1IER
  208   F836  8E 2E 10              STX P2IER
  209   F839  8E 4E 10              STX P3IER
  210   F83C  9A                    TXS               RESET THE STACK POINTER
  211   F83D  8E 18                 STX MAXPAX        ALL OUTGOING PACKETS WILL HAVE A MAX OF 128 BYTES
  212   F83F  A0 FF                 LDY #%11111111    PORTS 2 & 3A ARE OUTPUTS
  213   F841  8C 23 10              STY P2ADDR
  214   F844  8C 22 10              STY P2BDDR
  215   F847  8C 43 10              STY P3ADDR
  216   F84A  A2 19                 LDX #PRIMER       PRIME THE CNCT/DISC HANDSHAKE LINES
  217   F84C  8E 41 10              STX PORT3A
  218   F84F  8D 14 10              STA T1L           SET TIMER1 TO $4800 (1/100 SECOND)
  219   F852  A2 48                 LDX #$48
  220   F854  8E 15 10              STX T1MRH         THIS STARTS THE COUNTING
  221   F857  A2 19                 LDX #25           WANT TO COUNT 25 TIMEOUTS (1/4 SECOND)
  222   F859  86 4F                 STX TICK
  223   F85B              ;
  224   F85B  A2 93                 LDX #XADDR-1      ZERO OUT VARIABLES WHICH NEED TO BE RESET -
  225   F85D              ;                             (SEE SPACE ALLOCATION COMMANDS ABOVE)
  226   F85D              ;                              ( THE 'A' REG IS STILL SET TO ZERO)
  227   F85D  95 00       ZLOOP    STA 0,X
  228   F85F  CA                    DEX
  229   F860  10 FB                 BPL ZLOOP
  230   F862              ;
  231   F862              ;     (THE 'Y' REG IS STILL SET TO $FF)
  232   F862              ;
  233   F862  A2 02                 LDX #DUMATTY-QIN  SET THE RANGE FOR THE $FF INIT LOOP
  234   F864  94 66       FFLOOP   STY QIN,X         INITIALIZE THE VARIABLE
  235   F866  CA                    DEX
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
  235   F867  10 F8             BPL FFLOOP
  236   F869                ;
  237   F869  A2 16             LDX #BUFCNT     INITIALLY, BUFCNT ITEMS IN THE STACK
  238   F86B  86 3D             STX STKPTR      OFFSET OF TOP OF STACK + 1
  239   F86D  CA                DEX             INITIALIZE THE FREE BUFFER STACK
  240   F86E  8A           STACK1 TXA
  241   F86F  95 59             STA BUFSTK,X    PUT THE NUMBER OF EACH BUFFER IN THE STACK
  242   F871  CA                DEX
  243   F872  10 FA             BPL STACK1
  244   F874                ;
  245   F874                ; THE 'A' REG IS SET TO ZERO FROM ABOVE LOOP
  246   F874                ;
  247   F874  A2 15             LDX #BUFCNT-1   INITIALIZE THE FIRST BUFFER POINTER
  248   F876  95 95             STA HIPTR,X
  249   F878  A9 C0             LDA #<$00C0
  250   F87A  95 7F             STA LOPTR,X
  251   F87C  CA                DEX             SET THE REMAINING POINTERS, INITIALIZE LOOP CNTR
  252   F87D  A9 80             LDA #<$0180     SET BUFFER LOW AND HIGH PTRS IN LOPTR & HIPTR
  253   F87F  A0 01             LDY #>$0180     THE FIRST BUFFER IS AT ADDR $0180
  254   F881  95 7F       BUFFR1 STA LOPTR,X     SET THE LOW HALF OF THE ADDRESS
  255   F883  98                TYA
  256   F884  95 95             STA HIPTR,X     SET THE HIGH HALF OF THE ADDRESS
  257   F886  B5 7F             LDA LOPTR,X
  258   F888  18                CLC             INCR REG 'A' TO POINT TO THE NEXT BUFFER
  259   F889  69 80             ADC #BUFLEN
  260   F88B  90 01             BCC SKIP1
  261   F88D  C8                INY             IF THERE WAS A CARRY, INCR THE HIGH HALF
  262   F88E  CA           SKIP1 DEX
  263   F88F  10 F0             BPL BUFFR1
  264   F891                ;
  265   F891  AD 40 10          LDA PORT3B      STORE HOME ADDRESS IN RANDOM SEED
  266   F894  85 12             STA RANDU
  267   F896  A9 F0             LDA #240        XMIT FIRST STATUS MSG A MINUTE FROM NOW
  268   F898  85 1C             STA TSTAT
  269   F89A                ;
  270   F89A  A9 1F       DOCNCT LDA #HARADR    DO A CNCT/DISC CYCLE
  271   F89C  8D 41 10          STA PORT3A
  272   F89F  A9 02             LDA #CNCT       IS THE CONNECT FLAG SET? (G7 CONNECTED)
  273   F8A1  2C 40 10          BIT P3IFR
  274   F8A4  F0 F4             BEQ DOCNCT      NO, SO DO A CNCT/DISC CYCLE AGAIN
  275   F8A6                ;
  276   F8A6  A2 FF             LDX #$FF        AND SET CONECT TO WAITING FOR THE SYSTEM NAME
  277   F8A8                ;
  278   F8A8                ; THE FOLLOWING CODE WILL FORM A QUEUE OF SIGN-OFF MESSAGES TO EVERY POSSIBLE
  279   F8A8                ; HARRIS HIU IF THIS IS A POWER-UP CYCLE.  THIS IS NECESSARY BECAUSE OF
  280   F8A8                ; POSSIBLE LINKAGE TROUBLE CREATED BY A POWER FAILURE.  IF THIS IS NOT A
  281   F8A8                ; POWER-UP CYCLE, THE ONLY CHECK IS FOR A NORMAL SIGN-OFF MESSAGE TO
  282   F8A8                ; A CONNECTED DEVICE.
  283   F8A8                ;
  284   F8A8  A5 4D             LDA HARGU       TEST THE POWER-UP FLAG, IF RESET, = $57
  285   F8AA  C9 57             CMP #$57
  286   F8AC  F0 37             BEQ NOTPWR      RELIES ON ARBITRARY VALUE DURING POWER-UP
  287   F8AE  86 4E             STX CONECT      SET CONECT FLAG TO WAITING FOR SIGN-ON
  288   F8B0  A9 21             LDA #HARBAS     SET THE INITIAL HARRIS ADDRESS
  289   F8B2  85 4C             STA XADDR       SAVE IT IN XADDR FOR THE SIGN-OFF LOOP
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  290   F8B4              ;
  291   F8B4  A9 DE       SNDSOF  LDA #SOFMSG     SIGN OFF MESSAGE TYPE
  292   F8B6  20 E9 FE            JSR PCONST      CONSTRUCT THE PACKET
  293   F8B9  A9 17               LDA #QOUT+NEXT  ENQUEUE THE PACKET TO BE SENT
  294   F8BB  A4 0E               LDY CURBIU
  295   F8BD  20 43 FF            JSR ENQ
  296   F8C0              ;
  297   F8C0              ; THIS CODE LOOPS SENDING THE SIGN-OFF MESSAGES UNTIL ALL TERMINALS
  298   F8C0              ; HAVE BEEN ADDRESSED.
  299   F8C0              ;
  300   F8C0  A9 57               LDA #$57
  301   F8C2  C5 4D               CMP HARGO
  302   F8C4  F0 2E               BEQ NOSOF
  303   F8C6  E6 4C               INC XADDR       SET NEXT ADDRESS
  304   F8C8  A6 4C               LDX XADDR       TEST TO SEE IF DONE
  305   F8CA  E0 29               CPX #HARBAS+TRMCNT
  306   F8CC  D0 E6               BNE SNDSOF      NOT DONE, SO SEND ANOTHER SIGN-OFF MESSAGE
  307   F8CE  85 4D               STA HARGO       DONE, SO FLAG END OF POWER-UP CYCLE
  308   F8D0              ;
  309   F8D0              ; THE NEXT CODE GETS THE STANDARD HARRIS BIU ADDRESS FOR THIS BIU.  IT
  310   F8D0              ; IS BASED ON THE G7 BIU ADDRESS AND THE HARRIS BASE ADDRESS.
  311   F8D0              ;
  312   F8D0              ; NOW TRY TO SIGN-ONTO THE BASE HARRIS BIU
  313   F8D0              ;
  314   F8D0  AD 40 10    HARSON  LDA PORT3B      GET THIS BIU'S ADDRESS
  315   F8D3  29 0F               AND #$0F        TAKE ONLY THE LOW-ORDER 4 BITS
  316   F8D5  85 4C               STA XADDR       STORE IT TEMPORARILY
  317   F8D7  A9 21               LDA #HARBAS     GET THE BASE HARRIS ADDRESS
  318   F8D9  29 F0               AND #$F0        WANT ONLY THE HIGH-ORDER 4 BITS
  319   F8DB  05 4C               ORA XADDR       COMBINE THE TWO FOR THE TO ADDRESS
  320   F8DD  AA                  TAX             AND PUT THE ADDRESS IN X.
  321   F8DE  A9 08               LDA #8          SET UP TO USE PART OF INDEV CODE FOR SIGN-ON
  322   F8E0  20 2C FB            JSR SYSTMX
  323   F8E3  D0 18               BNE MLOOP0      ALWAYS BRANCH OUT
  324   F8E5              ;
  325   F8E5              ; TEST FOR A RESET SIGN-OFF MESSAGE
  326   F8E5              ;
  327   F8E5  A5 4E       NOTPWR  LDA CONECT      GET OLD CONECT FLAG
  328   F8E7  86 4E               STX CONECT
  329   F8E9              ;
  330   F8E9              ; TEST TO SEE IF WE WERE CONNECTED AND NEED TO SEND A SIGN-OFF MESSAGE.
  331   F8E9              ;
  332   F8E9  C9 01               CMP #01         WERE WE PREVIOUSLY CONNECTED?
  333   F8EB  D0 07               BNE NOSOF       IF NOT CONNECTED, DON'T SEND SIGN OFF MSG
  334   F8ED  A0 10               LDY #LINKT      PRINT "LINK TERMINATED"
  335   F8EF  20 63 FF            JSR PUTSTR
  336   F8F2  F0 C0               BEQ SNDSOF      ALWAYS BRANCH
  337   F8F4              ;
  338   F8F4  A0 00       NOSOF   LDY #00         CLEAR THE XMIT ADDR
  339   F8F6  84 4C               STY XADDR
  340   F8F8  A0 00               LDY #WHSYS
  341   F8FA  20 63 FF            JSR PUTSTR      PRINT "WHICH SYSTEM?"
  342   F8FD  20 9C FD            JSR INTBUF      SET UP A NETWORK INPUT BUFFER
  343   F900  58         MLOOP0   CLI             NOW READY TO PROCESS INTERRUPTS
```

```
CARD #  LOC   CODE        CARD. 10        20        30        40        50        60        70
  345   F901              ;
  346   F901              ; MLOOP IS THE MAIN LOOP. IT REPEATEDLY CALLS THE VARIOUS PROCESSING
  347   F901              ; ROUTINES UNTIL EITHER RESET IS HIT BY THE USER OR A NETWORK INTERRUPT
  348   F901              ; IS DETECTED.
  349   F901              ;
  350   F901  20 16 F9    MLOOP    JSR NET          HANDLE A MESSAGE TO THE NETWORK
  351   F904  20 81 F8             JSR OUTDEV       HANDLE A NETWORK MESSAGE FOR THE GRAPHIC 7
  352   F907  20 9C F0             JSR INTBUF       SEE IF A NETWORK INPUT BUFFER IS NECESSARY
  353   F90A  20 FA F9             JSR INDEV        HANDLE A GRAPHIC 7 MESSAGE FOR THE NETWORK
  354   F90D  20 93 FC             JSR TIMOUT       POLL THE TIMER
  355   F910  20 9E FC             JSR CKTOUT       CHECK FOR ANY TIMEOUTS
  356   F913  4C 01 F9             JMP MLOOP        CONTINUE LOOPING
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  358   F916              ;
  359   F916              ; NET IS CALLED TO OUTPUT MESSAGES ONTO THE NETWORK. IF IT
  360   F916              ; CAN'T FOR ANY REASON, IT RETURNS AND TRIES AGAIN THE NEXT TIME
  361   F916              ; IT IS CALLED.
  362   F916              ; GRAPHIC 7 AND HARRIS VERSIONS OF NET SHOULD BE IDENTICAL.
  363   F916              ;
  364   F916  A6 67       NET       LDX OUUT        IS THERE ANYTHING WAITING TO GO OUT?
  365   F918  30 4D               BMI RNET        IF NOTHING, RETURN
  366   F91A              ;
  367   F91A              ; A CHECK IS MADE TO SEE IF WE TRIED TO TRANSMIT SOMETHING ON THE NETWORK
  368   F91A              ; EARLIER AND COULD NOT.  THERE ARE TWO POSSIBLE REASONS FOR THE FAILURE: A
  369   F91A              ; COLLISION WITH SOMEONE ELSE'S TRANSMISSION OR AN 'FF' ACK FROM THE ADDRESSEE
  370   F91A              ; INDICATING HIS BUFFERS ARE FULL.
  371   F91A              ;
  372   F91A              ; IF A COLLISION OCCURRED THEN A RANDOM NUMBER IS LOADED INTO RNDCNT.
  373   F91A              ;
  374   F91A              ; IF AN 'FF' ACK WAS RECEIVED THEN 255 WAS LOADED INTO NOXMIT.
  375   F91A              ;
  376   F91A              ; THE FOLLOWING LOOPS ALLOW BOTH PROBLEMS TO CLEAR.
  377   F91A              ;
  378   F91A  A5 3E       LWAIT     LDA NOXMIT      WERE WE ASKED TO STOP XMITTING FOR A WHILE?
  379   F91C  F0 06               BEQ SWAIT1      NO, SO CHECK FOR RANDOM BACKOFF
  380   F91E  A9 32               LDA #50         THIS GIVES US A 0.2 MILLISECOND WAIT
  381   F920  C6 3E               DEC NOXMIT      MARK THAT WE ARE WAITING 0.2 MS
  382   F922  85 13               STA RNDCNT      STORE THE CONSTANT FOR 0.2 MS
  383   F924  A5 13       SWAIT1    LDA RNDCNT      IS RANDOM BACKOFF OR LONG WAITING ACTIVE?
  384   F926  F0 06               BEQ PREPAR      NO
  385   F928  C6 13       SWAIT2    DEC RNDCNT      COUNTDOWN RANDOM WAIT OR 0.2 MS WAIT
  386   F92A  D0 FC               BNE SWAIT2
  387   F92C  F0 EC               BEQ LWAIT       SEE IF WE STILL HAVE THE LONG WAIT ACTIVE.
  388   F92E              ;
  389   F92E              ; PREPARE TO SEND PACKET BY CHECKING TO SEE IF THE NET IS FREE.  ALSO LOAD THE
  390   F92E              ; PACKET HEADER WHILE WAITING.
  391   F92E              ;
  392   F92E  85 38       PREPAR    STA PARITY      ZERO OUT VERTICAL PARITY('A'=0 FROM WAIT LOOP)
  393   F930  A9 FF               LDA #$FF        SET INTERRUPT DETECTOR
  394   F932  85 40               STA INTFLG
  395   F934  AD 00 0C            LDA NUARTS      IS THE NETWORK BUSY?
  396   F937  29 04               AND #%00000100
  397   F939  F0 27               BEQ NETBSY      IF BUSY, THEN WAIT FOR LATER
  398   F93B              ;
  399   F93B              ; NET NOT BUSY NOW BUT COULD BE WAITING FOR AN ACK.  SET UP HEADER AND TRY
  400   F93B              ;   AGAIN IN 100US.
  401   F93B              ;
  402   F93B  B4 7F               LDY LOPTR,X     GET THE POINTER TO THE BUFFER TO SEND
  403   F93D  84 06               STY OUTPTR
  404   F93F  B4 95               LDY HIPTR,X
  405   F941  84 07               STY OUTPTR+1
  406   F943  A5 34               LDA SNUM        SAVE THE SEQUENCE # IN THE PACKET
  407   F945  A0 04               LDY #04
  408   F947  91 06               STA (OUTPTR),Y
  409   F949  A5 39               LDA NTPRX       SAVE THE RETRANSMISSION COUNT IN THE PACKET
  410   F94B  A0 06               LDY #06
  411   F94D  91 06               STA (OUTPTR),Y
  412   F94F  C8                  INY
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
  413   F950  B1 06             LDA (OUTPTR),Y  GET THE PACKET LENGTH
  414   F952  85 0A             STA OUTBC       SAVE IT FOR THE OUTPUT LOOP
  415   F954  A0 19             LDY #25         FINISH WAITING FOR 100 US AFTER PREPAR
  416   F956  88         WAIT100 DEY
  417   F957  10 FD             BPL WAIT100
  418   F959  A5 40             LDA INTFLG      GET ZERO, IF INTERRUPT OCCURRED, ELSE GET ONES
  419   F95B  2D 00 0C          AND NUARTS      GET CURRENT NETWORK STATUS
  420   F95E  29 04             AND #%00000100
  421   F960  D0 06             BNE TRNSMT      IF IT IS NOT BUSY, OUTPUT A MESSAGE
  422   F962  A2 36      NETBSY LDX #NMWAIT     INCR # OF TIMES WE HAVE TO WAIT FOR THE NET
  423   F964  20 2C FF          JSR SFINC
  424   F967  60         NRET   RTS
  425   F968                    ;
  426   F968                    ; TRANSMIT THE MESSAGE.  TURN ON XMIT KEY.  THIS SHOULD HOLD THE NETWORK FOR
  427   F968                    ; US, BUT KEEP CHECKING FOR COLLISIONS BY MONITORING THE XMIT FLAG.
  428   F968                    ;
  429   F968  A9 98      TRNSMT LDA #%10011000  TURN ON XMIT KEY
  430   F96A  8D 00 0C          STA NUARTS
  431   F96D  85 14             STA XMIT        SET XMIT FLAG TO NONZERO VALUE
  432   F96F                    ; THE 'Y' REG IS STILL = $FF FROM THE WAIT100 LOOP.
  433   F96F  C8         TXLOOP INY
  434   F970  A5 14      TXBEMP LDA XMIT        HAS XMIT FLAG BEEN RESET? (COLLISION?)
  435   F972  D0 06             BNE NTXCOL      NO, SO KEEP GOING
  436   F974  A2 2C      TXCOL  LDX #NMCOL      YES, SO INCREMENT NUMBER OF COLLISIONS
  437   F976  20 2C FF          JSR SFINC
  438   F979  60                RTS
  439   F97A  AD 00 0C   NTXCOL LDA NUARTS      IS XMIT BUFFER EMPTY?
  440   F97D  29 02             AND #%00000010
  441   F97F  F0 EF             BEQ TXBEMP
  442   F981  B1 06             LDA (OUTPTR),Y  WRITE WORD TO NETWORK
  443   F983  8D 01 0C          STA NUARTD
  444   F986  45 38             EOR PARITY      ACCUMULATE THE PARITY
  445   F988  85 38             STA PARITY
  446   F98A  C4 08             CPY OUTBC       ARE WE DONE?
  447   F98C  D0 E1             BNE TXLOOP      NO, SO LOOP SOME MORE
  448   F98E                    ;
  449   F98E  A5 14      TXPAR  LDA XMIT        HAS XMIT BEEN TURNED OFF BY NMI?
  450   F990  F0 E2             BEQ TXCOL       IF SO, SIGNAL COLLISION
  451   F992  AD 00 0C          LDA NUARTS      OUTPUT THE PARITY TO THE NETWORK
  452   F995  29 02             AND #%00000010  IS THE XMIT BUFFER EMPTY?
  453   F997  F0 F5             BEQ TXPAR        NO, SO KEEP TRYING
  454   F999  A5 38             LDA  ARITY       YES, SO SEND VERTICAL PARITY
  455   F99B  8D 01 0C          ,A NUARTD
  456   F99E  A2 1A             LDX #26         DELAY TURNING OFF THE KEY
  457   F9A0  CA         HOLDTX DEX
  458   F9A1  10 FD             BPL HOLDTX
  459   F9A3  A9 58             LDA #%01011000  TURN OFF XMIT KEY (THIS MUST OCCUR
  460   F9A5  8D 00 0C          STA NUARTS      AT LEAST 75 USEC AFTER LAST BYTE)
  461   F9A8                    ;
  462   F9A8  A5 14      KEYNOF LDA XMIT        WAIT FOR THE KEY TO TURN OFF
  463   F9AA  D0 FC             BNE KEYNOF
  464   F9AC  78                SEI             NO INTERRUPTS FOR A WHILE
  465   F9AD                    ;
  466   F9AD                    ; REG 'A' IS = 0 FROM KEYNOF LOOP ABOVE
  467   F9AD                    ;
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  468   F9AD  A8                 TAY
  469   F9AE  B1 06               LDA (OUTPTR),Y  GET THE TO ADDRESS
  470   F9B0  85 08               STA OUTBC       SAVE IT IN OUTBC
  471   F9B2  E6 14               INC XMIT        MAKE XMIT NON-ZERO
  472   F9B4  A2 11               LDX #17         LOOP FOR 100 US
  473   F9B6  AD 00 0C  AKLOOP    LDA NUARTS      WAIT FOR THE ACK TO COME BACK
  474   F9B9  30 1A               BMI ACKCK       A CHARACTER HAS ARRIVED
  475   F9BB  CA                  DEX             KEEP ON WAITING
  476   F9BC  10 F8               BPL AKLOOP
  477   F9BE                ;
  478   F9BE                ; 'Y' REG = 0 FROM KEYNOF ABOVE
  479   F9BE                ;
  480   F9BE  84 14         BADACK    STY XMIT          NOTHING ARRIVED, SO TURN OFF XMIT FLAG
  481   F9C0  58                      CLI               RESTORE INTERRUPTS
  482   F9C1  A2 2A                   LDX #NMRXMT       COUNT # OF TIMES NO ACK
  483   F9C3  20 2C FF                JSR SFINC
  484   F9C6  E6 39                   INC NTPRX         INCR # OF TIMES THIS MSG IS TRANSMITTED
  485   F9C8  A6 39                   LDX NTPRX         SENT 127 TIMES? (LIMIT UPPED CHECKOUT)
  486   F9CA  E0 7F                   CPX #127
  487   F9CC  90 28                   BCC NETRET        NO, SO TRY IT AGAIN LATER
  488   F9CE  A2 2E                   LDX #NMDISC       YES, SO INCR THE # OF DISCARDED MESSAGES
  489   F9D0  20 2C FF                JSR SFINC
  490   F9D3  D0 15                   BNE NETFRE        AND FREE UP THE PACKET BUFFER (ALWAYS BRANCH)
  491   F9D5                ;
  492   F9D5                ; CHECK FOR GOOD ACKNOWLEDGEMENT OR A FULL BUFFER NACK(FF).  THEN TAKE
  493   F9D5                ; PROPER ACTION.
  494   F9D5                ;
  495   F9D5  AD 01 0C  ACKCK     LDA NUARTD        GET THE CHARACTER THAT ARRIVED
  496   F9D8  C5 08               CMP OUTBC         IS IT THE ACK?
  497   F9DA  F0 08               BEQ ACKOK         YES
  498   F9DC  C9 FF               CMP #$FF          IS IT A FLAG THAT THE RECEIVER IS OVERRUN?
  499   F9DE  D0 DE               BNE BADACK        NO
  500   F9E0  85 3E               STA NOXMIT        YES, SO DON'T XMIT FOR 50 MILLISECONDS
  501   F9E2  F0 DA               BEQ BADACK        ALWAYS BRANCH
  502   F9E4                ;
  503   F9E4  58            ACKOK     CLI               CLEAR INTERRUPTS
  504   F9E5  A2 28                   LDX #NMXMT        INCR THE # OF TRANSMITTED MSGS
  505   F9E7  20 2C FF                JSR SFINC
  506   F9EA                ;
  507   F9EA                ; FREE THE BUFFER THAT WAS JUST PROCESSED OR COULDN'T BE SENT.
  508   F9EA                ;
  509   F9EA  A2 17         NETFRE    LDX #OUT-NEXT     FREE THE BUFFER THAT WAS SENT
  510   F9EC  20 51 FF                JSR DQ
  511   F9EF  A9 00                   LDA #00           RE-INITIALIZE # OF TIMES CURRENT MSG XMITTED
  512   F9F1  85 39                   STA NTPRX
  513   F9F3  E6 3A                   INC SQNUM         SET THE SEQUENCE # FOR THE NEXT PACKET
  514   F9F5  A5 14         ANDONE    LDA XMIT          WAIT FOR THE KEY TO TURN OFF
  515   F9F7  D0 FC                   BNE ANDONE
  516   F9F9  60            NETRET    RTS               AND RETURN
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  518   F9FA              ;
  519   F9FA              ; INDEV IS CALLED TO POLL FOR CHARACTERS FROM THE GRAPHIC 7.
  520   F9FA              ; TWO MODES ARE ASSUMED: TTY MODE AND DMA MODE. (DMATTY=0 AND 1 RESPECTFULLY)
  521   F9FA              ;     TTY MODE MEANS CHARACTERS ARE IN ASCII FORMAT WITH A LEADING BYTE OF
  522   F9FA              ;     ZERO'S IN EACH WORD.  CONTROL CHARACTERS SHOULD RESULT IN THE BUFFERED
  523   F9FA              ;     PACKET BEING SENT.
  524   F9FA              ;
  525   F9FA              ;     DMA MODE MEANS BINARY FORMAT IS USED AND CONTROL CHARACTERS MAY BE
  526   F9FA              ;     EMBEDDED .  THE FINAL PACKET WILL BE TRANSMITTED WHENEVER SUFFICIENT TIME
  527   F9FA              ;     HAS ELAPSED TO INDICATE THE END OF THE DATA STREAM, (APPROX 600 USECS).
  528   F9FA              ;
  529   F9FA  A5 4E       INDEV   LDA CONECT      ARE WE CONNECTED TO ANOTHER BIU?
  530   F9FC  30 0E               BMI GETCH1      NO, SO JUST TRY TO GET A CHAR
  531   F9FE  F0 0C               BEQ GETCH1
  532   FA00  A5 0F               LDA INSET       YES, SO DO WE HAVE AN INPUT BUFFER SET UP YET?
  533   FA02  D0 20               BNE GETCH       YES, SO TRY TO GET A CHAR
  534   FA04  A9 02               LDA #DATMSG+TTY NO, SO TRY TO ALLOCATE A DATA BUFFER
  535   FA06  20 E9 FE            JSR PCONST
  536   FA09  10 07               BPL ABUF        IF COULD ALLOCATE THE BUFFER, CONTINUE
  537   FA0B  60          IRET    RTS             IF COULDN'T, RETURN AND TRY LATER
  538   FA0C              ;
  539   FA0C              ;  IF THIS BIU IS NOT CONNECTED TO ANOTHER BIU, FORCE NULL MODE PROCESSING
  540   FA0C              ;  OF DATA PRESENTED AT THE INTERFACE.
  541   FA0C              ;
  542   FA0C  A9 FF       GETCH1  LDA #$FF        SET DMATTY TO -1
  543   FA0E  85 66               STA DMATTY
  544   FA10  D0 12               BNE GETCH       ALWAYS BRANCH
  545   FA12              ;
  546   FA12              ; SET UP THE BUFFER POINTERS
  547   FA12              ;
  548   FA12  A4 0A       ABUF    LDY BIUPTR      SET THE INPUT BUFFER POINTER
  549   FA14  84 03               STY INPTR
  550   FA16  A4 0B               LDY BIUPTR+1
  551   FA18  84 04               STY INPTR+1
  552   FA1A  A4 0E               LDY CURBIU      THE BUFFER NUMBER
  553   FA1C  84 00               STY CURDEV
  554   FA1E  E6 0F               INC INSET       NOW SET UP FOR INPUT
  555   FA20  A0 08               LDY #8          OFFSET INTO THE BUFFER OF THE FIRST CHAR
  556   FA22  84 05               STY INHC
  557   FA24              ;
  558   FA24              ; GETCH TRIES TO GET A WORD. IF IT IS SUCCESSFUL, THE WORD IS PASSED TO
  559   FA24              ; THE PROPER CODE TO HANDLE IT.
  560   FA24              ;
  561   FA24  A4 05       GETCH   LDY INHC        SET Y IN CASE IT IS NECESSARY TO OUTPUT BUFFER
  562   FA26  A9 02               LDA #INR        IS THERE ANY DATA READY FROM THE DEVICE?
  563   FA28  2C 1D 10            BIT PIIFR
  564   FA2B  D0 22               BNE GETDAT      YES, SO GO GET IT
  565   FA2D              ;
  566   FA2D              ; DMATTY IS INITIALIZED TO -1 IN THE FFLOOP.  IT IS SET TO 1 ON THE FIRST
  567   FA2D              ; PASS THROUGH GETDAT TO ASSUME A DMA OPERATION.  IT IS SET TO 0 IF TIMING
  568   FA2D              ; INDICATES IT IS NOT A DMA OPERATION OR A ONE WORD DMA OP.  RESET TO -1
  569   FA2D              ; BY CODE IN INDONE.
  570   FA2D              ;
  571   FA2D  A5 66               LDA DMATTY      NO, ARE WE IN THE MIDDLE OF AN OPERATION?
  572   FA2F  30 DA               BMI IRET        NO, SO RETURN
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  573   FA31  F0 0C              BEQ TTYOP       TTY OPERATIO
  574   FA33              ;
  575   FA33              ; DMATM IS A TIMER SET DURING DMA OPS TO TIME HOW LONG IT TAKES FOR THE
  576   FA33              ; NEXT WORD TO ARRIVE.  IT IS SET EACH PASS THROUGH GETDAT AND CHECKED BELOW.
  577   FA33              ;
  578   FA33  C6 42             DEC DMATM       DMA OPERATION; HAS THE DMA TIMER GONE OFF
  579   FA35  D0 ED             BNE GETCH       NO, SO CONTINUE TO WAIT
  580   FA37  C6 68             DEC DMATTY      YES, SO MUST BE DONE OR A TTY OPERATION
  581   FA39              ;
  582   FA39              ; NOW CHECK TO SEE IF MORE THAN ONE WORD CAME THROUGH AS DMA.  IF 'Y' IS
  583   FA39              ; GREATER THAN OR EQUAL TO 11 THEN THIS WAS PROBABLY A DMA OP AND WE SHOULD
  584   FA39              ; SEND THE BUFFERED DATA.
  585   FA39              ;
  586   FA39  C0 0B             CPY #11         IF DMA, MORE THAN ONE WORD STORED ALREADY?
  587   FA3B  10 5A             BPL INDONE      YES, SO SEND THE PACKET
  588   FA3D  E6 43             INC TTYTM       NO, SO PROBABLY TTY; SET THE TTY TIMER FLAG
  589   FA3F              ;
  590   FA3F              ;  NOT MORE THAN ONE WORD, SO PROBABLY A TTY OP.  SET THE TTY TIMER FLAG
  591   FA3F              ;  TO CLEAR ONE WORD DMA OPS OR TTY OPS NOT TERMINATED WITH CARRIAGE CONTROL.
  592   FA3F              ;
  593   FA3F  A5 43       TTYOP   LDA TTYTM       HAS THE TTY TIMER FLAG BEEN RESET?
  594   FA41  F0 54             BEQ INDONE      YES, SO SEND THE PACKET. (CATCHES ONE WORD DMA OPS)
  595   FA43  A5 41             LDA BYTLOW      NO, SO MUST BE TTY OPERATION; GET THE LAST BYTE
  596   FA45  C9 20             CMP #$20        WAS THE LAST BYTE A CONTROL CHARACTER?
  597   FA47  90 4E             BCC INDONE      YES, SO SEND THE PACKET
  598   FA49  C9 7F             CMP #$7F        WAS IT A BACKSPACE/DELETE CHAR?
  599   FA4B  F0 4A             BEQ INDONE      YES, SO SEND THE PACKET
  600   FA4D  D0 3C             BNE IRET        NO, SO GET ANOTHER CHAR
  601   FA4F              ;
  602   FA4F  A6 68       GETDAT  LDX DMATTY      WHAT MODE OF OPERATION IS IN PROGRESS?
  603   FA51  F0 0A             BEQ TTYMOD      TTY MODE
  604   FA53  10 04             BPL DMAMOD      DMA MODE
  605   FA55              ;
  606   FA55              ; IF DMATTY EQUALS -1 WE ASSUME THIS IS THE START OF A DMA OPERATION.
  607   FA55              ;
  608   FA55  A2 01             LDX #1
  609   FA57  86 68             STX DMATTY      NEITHER MODE SO ASSUME DMA
  610   FA59  A2 20       DMAMOD  LDX #$20       SET THE DMA TIMER
  611   FA5B  86 42             STX DMATM
  612   FA5D              ;
  613   FA5D              ; THE TTY TIMER IS USED TO KEEP A ONE WORD DMA PACKET FROM BEING DELAYED
  614   FA5D              ; FOR TOO LONG.  IT ASSUMES THAT TELETYPE ENTRIES WILL ARRIVE AT SOMETHING
  615   FA5D              ; LESS THAN 1 SECOND INTERVALS.  IF NOTHING ARRIVES AFTER ONE SECOND THE
  616   FA5D              ; PACKET IS SENT.
  617   FA5D              ;
  618   FA5D  A2 25       TTYMOD  LDX #TETTY      SET THE TTY TIMER
  619   FA5F  A9 04             LDA #4          TIMER IS SET FOR ONE SECOND
  620   FA61  20 1A FF          JSR STIMER
  621   FA64  AD 10 10          LDA PORT1B      GET THE HIGH ORDER BYTE
  622   FA67  49 FF             EOR #$FF        TAKE THE 1'S COMPLEMENT OF THE DATA
  623   FA69  48                PHA             STORE IT TEMPORARILY
  624   FA6A              ;
  625   FA6A              ; NOTE: LOADING FROM PORT 1A WILL TRIGGER AN AUTOMATIC HARDWARE HANDSHAKE
  626   FA6A              ;       SIGNAL FROM THE 6522 CHIP.  THIS IS THE EQUIVALANT OF THE G7 ICTL
  627   FA6A              ;       (HARRIS DATO) SIGNAL.
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  628   FA6A              ;
  629   FA6A  AD 11 10            LDA PORTIA      GET THE LOW ORDER BYTE AND TRIGGER HANDSHAKING
  630   FA6D  49 FF               EOR #$FF        TAKE THE 1'S COMPLEMENT OF THE DATA
  631   FA6F  85 41               STA BYTLOW      SAVE IT FOR LATER
  632   FA71  68                  PLA             RESTORE THE HIGH ORDER BYTE
  633   FA72  A4 17               LDY WAIT        ARE WE WAITING FOR A SIGN-ON ACK?
  634   FA74  D0 95               BNE IRET        YES, SO IGNORE THE DATA
  635   FA76              ;
  636   FA76              ; CONNECT WILL TELL US WHICH ROUTINE BELOW IS TO PROCESS THE INPUT.
  637   FA76              ;
  638   FA76  A6 4E               LDX CONECT      GET THE CURRENT STATE OF DEVICE INPUT
  639   FA78  30 3A               BMI GETRPY      IF WAITING FOR REPLY TO WHICH SYSTEM REQUEST
  640   FA7A  D0 03               BNE CHARS       IF ADDING CHARS TO OUTPUT BUFFER
  641   FA7C  4C 47 FB            JMP GETYN       IF WAITING FOR REPLY TO LINK REQUEST (Y OR N)
```

```
CARD # LOC   CODE       CARD 10        20        30        40        50        60        70
  643 FA7F             ;
  644 FA7F             ; ADD THE CHARACTER TO THE OUTPUT BUFFER AND QUEUE THE BUFFER WHEN
  645 FA7F             ; IT IS FULL.
  646 FA7F             ;
  647 FA7F A4 05       CHARS   LDY INBC        THE PLACE TO ADD THE CHAR TO THE BUFFER
  648 FA81 91 03               STA (INPTR),Y   STORE IT IN THE BUFFER
  649 FA83 C4 18               CPY MAXPAX      IS THE BUFFER FULL?
  650 FA85 F0 1A               BEQ INDON1      YES
  651 FA87 E6 05               INC INBC
  652 FA89 20 90 FA            JSR BYTES       WAS THIS THE FIRST OR SECOND BYTE?
  653 FA8C B0 F1               BCS CHARS       FIRST, SO GO GET SECOND
  654 FA8E 90 94               BCC GETCH       SECOND, SO GO GET ANOTHER WORD
  655 FA90             ;
  656 FA90             ; BYTES IS USED TO DETERMINE WHICH HALF OF A WORD WAS PROCESSED.  THE CARRY
  657 FA90             ; BIT IS USED TO INDICATE AN ODD OR EVEN BYTE COUNT.  (ODD=>CS; EVEN=>CC)
  658 FA90             ;
  659 FA90 A5 05       BYTES   LDA INBC        FIRST BYTE IS INDICATED BY AN ODD INBC COUNT
  660 FA92 18                  CLC
  661 FA93 6A                  ROR A           ROTATE THE LOW ORDER BIT INTO THE CARRY
  662 FA94 A5 41               LDA BYTLOW      SET UP THE SECOND BYTE IN A REG
  663 FA96 60          BYTRET  RTS
  664 FA97             ;
  665 FA97             ; INDONE IS USED TO SEND PACKETS THAT HAVE BEEN FORMED DURING TERMINAL I/O
  666 FA97             ; IT CHECKS FOR NULL PACKETS (INBC<8) AND RESETS DMATTY & TTYTM FLAGS IF NOT
  667 FA97             ; IN THE MIDDLE OF A DMA TRANSFER.
  668 FA97             ;
  669 FA97 A2 FF       INDONE  LDX #$FF        RESET DMATTY FLAG
  670 FA99 86 68               STX DMATTY
  671 FA9B E8                  INX             RESET THE TTY TIMER FLAG
  672 FA9C 86 43               STX TTYTM
  673 FA9E 88                  DEY             SET PACKET LENGTH
  674 FA9F 98          INDON1  TYA             AND PUT VALUE IN A REGISTER
  675 FAA0 C9 07               CMP #7          IS IT A NULL PACKET?
  676 FAA2 F0 F2               BEQ BYTRET      YES, SO IGNORE IT.
  677 FAA4 A0 07               LDY #07
  678 FAA6 91 03               STA (INPTR),Y
  679 FAA8 A9 17               LDA #OUT+NEXT   ENQUEUE THE PACKET TO BE SENT
  680 FAAA A4 0D               LDY CURDEV
  681 FAAC 20 43 FF            JSR END
  682 FAAF C6 0F               DEC INSET       NO LONGER SET UP
  683 FAB1 4C FA F9            JMP INDEV       TRY TO GET THE NEXT BUFFER & MORE DATA IF AVAILABLE
  684 FAB4             ;
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
   686  FAB4              ;
   687  FAB4              ; GETRPY ACCUMULATES THE RESPONSE TO THE WHICH SYSTEM MESSAGE.
   688  FAB4              ; WHEN THE RESPONSE IS COMPLETE, IT IS CHECKED WITH A SET OF TABLE ENTRIES
   689  FAB4              ; AT THE END OF THE PROGRAM.
   690  FAB4              ;
   691  FAB4  A5 41       GETRPY  LDA BYTLOW      GET THE LOW ORDER BYTE
   692  FAB6  29 5F               AND #$5F        GET RID OF PARITY AND UPPER/LOWER CASE
   693  FAB8  A6 05               LDX INBC        THE OFFSET IN STRING TO PUT THE CHAR
   694  FABA  95 47               STA STRING,X
   695  FABC  C9 0D               CMP #$0D        WAS THE CHAR A CR?
   696  FABE  F0 09               BEQ GLF         YES, SO OUTPUT A LF
   697  FAC0  E0 04               CPX #$04        WAS IT THE FIFTH CHAR?
   698  FAC2  F0 09               BEQ GCRLF       YES, SO OUTPUT A CRLF
   699  FAC4  E6 05               INC INBC        NEITHER OF THE ABOVE, SO WAIT FOR MORE INPUT
   700  FAC6  4C 0C FA            JMP GETCH1      GO GET ANOTHER WORD
   701  FAC9              ;
   702  FAC9  A0 0E       GLF     LDY #LF
   703  FACB  D0 02               BNE GPUT        ALWAYS BRANCH
   704  FACD  A0 0D       GCRLF   LDY #CRLF
   705  FACF  20 63 FF    GPUT    JSR PUTSTR      PRINT THE CHARS
   706  FAD2              ;
   707  FAD2              ; COMPARE THE INPUT STRING TO DEFINED SYSTEM NAMES IN THE TABLE
   708  FAD2              ;
   709  FAD2  A0 00               LDY #00         SEE IF RESPONSE IS STORED IN THE TABLE
   710  FAD4  A2 00       NEXTEN  LDX #00          (CHARACTER POSITION COUNTER)
   711  FAD6  B5 47       NEXTCH  LDA STRING,X
   712  FAD8  D9 75 FF            CMP TABLE,Y
   713  FADB  D0 08               BNE NMATCH      NO MATCH, SO TRY NEXT ENTRY
   714  FADD  E0 02               CPX #02         IS THIS THIRD CHARACTER?
   715  FADF  F0 22               BEQ MATCH       YES, WE HAVE A MATCH
   716  FAE1  E8                 INX
   717  FAE2  C8                 INY
   718  FAE3  D0 F1               BNE NEXTCH      TRY THE NEXT CHAR (ALWAYS BRANCH)
   719  FAE5  98          NMATCH  TYA
   720  FAE6  86 15               STX TOKEN       NO MATCH, SO TRY THE NEXT SYSTEM
   721  FAE8  38                 SEC
   722  FAE9  E5 15               SBC TOKEN
   723  FAEB  18                 CLC
   724  FAEC  69 04               ADC #04         FOUR BYTES PER ENTRY
   725  FAEE  C9 10               CMP #16         MORE SYSTEMS TO TRY (FOR 4 ENTRIES)?
   726  FAF0  F0 03               BEQ ASKAGN      NO, SO ASK HIM AGAIN
   727  FAF2  A8                 TAY
   728  FAF3  D0 DF               BNE NEXTEN      TRY THE NEXT ENTRY IN TABLE (ALWAYS BRANCH)
   729  FAF5              ;
   730  FAF5              ; ASKAGN RESETS SOME SYSTEM FLAGS AND REPRINTS THE 'WHICH SYSTEM?' MESSAGE
   731  FAF5              ; AT THE TERMINAL.  THIS INDICATES THE ENTRY WAS NOT IN THE TABLE.
   732  FAF5              ;
   733  FAF5  A2 00       ASKAGN  LDX #00
   734  FAF7  86 05               STX INBC        RESET THE CHARACTER POINTER
   735  FAF9  86 17               STX WAIT        NOT WAITING FOR A SIGN-ON ACK YET
   736  FAFB  86 4C               STX XADDR       NOT TALKING TO ANYONE
   737  FAFD  A0 00               LDY #WHSYS      ASK WHICH SYSTEM?
   738  FAFF  20 63 FF            JSR PUTSTR
   739  FB02  60                  RTS             AND RETURN
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  741   FB03              ;
  742   FB03              ; MATCH SETS UP SOME DEVICE PARAMETERS ACCORDING TO THE VALUES IN THE TABLE.
  743   FB03              ; IF THIS WAS A TERMINAL REQUEST, IT SETS THE WAIT TIMER FOR 10 SECONDS,
  744   FB03              ; IF A CPU DEVICE OTHER THAN A HARRIS BIU IT SETS THE TIMER FOR 2 SECONDS.
  745   FB03              ; HARRIS BIU SIGN-ONS WAIT FOR 1/2 SECOND.  THE LINK REQUEST MESSAGE IS THEN
  746   FB03              ; QUEUED FOR TRANSMISION AND WE START TO AWAIT THE RESPONSE.
  747   FB03              ;
  748   FB03  C0 08       MATCH   CPY #08           A MATCH WAS FOUND IN THE TABLE
  749   FB05  90 0F               BCC TERMX         IF "TERMX" OR "SANDX", GET THE TERMINAL ADDRESS
  750   FB07  C0 0A               CPY #$0A          IF "CAP", USE SIGN-ON LOOP
  751   FB09  D0 03               BNE CONMAT
  752   FB0B  4C D0 FB            JMP HARSON
  753   FB0E  C8          CONMAT  INY               ELSE GET THE ADDRESS FROM THE TABLE
  754   FB0F  BE 73 FF            LDX TABLE,Y       ACC = DA
  755   FB12  A9 05               LDA #5            WAIT 2 SECONDS FOR A REPLY TO THE SIGN-ON
  756   FB14  D0 16               BNE SYSTMX        ALWAYS BRANCH
  757   FB16              ;
  758   FB16  A2 04       TERMX   LDX #04           GET DA FROM "X" IN "TERMX" OR "SANDX"
  759   FB18  B5 47               LDA STRING,X
  760   FB1A  C0 04               CPY #4
  761   FB1C  F0 08               BEQ CONTRM        "TERMX" SO CONTINUE
  762   FB1E  29 0F               AND #$0F          TAKE THE LOW-ORDER 4 BITS FROM X
  763   FB20  85 4C               STA XADDR         STORE IT AWAY TEMPORARILY
  764   FB22  AD 40 10            LDA PORT38        GET THE HIGH-ORDER BITS FROM THIS BIU'S ADDRESS
  765   FB25  29 F0               AND #$F0
  766   FB27  05 4C               ORA XADDR         AND COMBINE FOR SIGN-ON ADDRESS
  767   FB29  AA          CONTRM  TAX               PUT SIGN-ON ADDRESS IN X
  768   FB2A  A9 28               LDA #40           WAIT 10 SECONDS FOR A REPLY TO THE SIGN-ON
  769   FB2C  86 4C       SYSTMX  STX XADDR         SAVE THE DESTINATION ADDR
  770   FB2E  A2 1F               LDX #TSACK
  771   FB30  20 1A FF            JSR STIMER        SET THE TIMER TO WAIT FOR THE REPLY
  772   FB33  A9 E2               LDA #SONMSG+TTY   CONNECTION MSG FOR ASCII TERMINALS
  773   FB35  20 E9 FE            JSR PCONST        CONSTRUCT THE PACKET
  774   FB38  10 03               BPL QIT           IF A BUFFER WAS ALLOCATED
  775   FB3A  4C 00 F8            JMP RESET         IF NOT THERE, SOMETHING IS WRONG SO RESET.
  776   FB3D              ;
  777   FB3D  A9 17       QIT     LDA #QOUT+NEXT    ENQUEUE THE BUFFER TO BE XMITTED
  778   FB3F  A4 0E               LDY CURBIU
  779   FB41  20 43 FF            JSR ENQ
  780   FB44  E6 17               INC WAIT          REMEMBER THAT WE ARE WAITING FOR A REPLY
  781   FB46  60          INRET   RTS               AND RETURN
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  783   FB47              ;
  784   FB47              ; GETYN IS USED TO GET THE REPLY TO A LINK REQUEST (Y OR N).
  785   FB47              ;
  786   FB47  29 5F       GETYN     AND #$5F          GET RID OF PARITY AND UPPER/LOWER CASE
  787   FB49  D0 05                 BNE NOZERO        IGNORE 0'S IN TTY MODE
  788   FB4B  20 90 FA              JSR BYTES         FIRST OR SECOND BYTE?
  789   FB4E  B0 F7                 BCS GETYN         FIRST, SO GET SECOND
  790   FB50  C9 59       NOZERO    CMP #'Y           DID THE USER TYPE YES?
  791   FB52  F0 0A                 BEQ YES           YES
  792   FB54  C6 4E                 DEC CONECT        NO, SO AGAIN WAITING FOR WHICH SYSTEM
  793   FB56  A0 0D                 LDY #CRLF         PRINT CRLF
  794   FB58  20 63 FF              JSR PUTSTR
  795   FB5B  4C F5 FA              JMP ASKAGN        AND ASK HIM WHAT HE WANTS
  796   FB5E              ;
  797   FB5E              ; WHEN LINKING TO ANOTHER TERMINAL, SET MAXIMUM PACKET LENGTH, SEND AN
  798   FB5E              ; ACKNOWLEDGMENT AND TYPE LINK ACCEPTED AT THE TERMINAL.
  799   FB5E              ;
  800   FB5E  A9 DF       YES       LDA #SACMSG       SEND A SIGN-ON ACK
  801   FB60  20 E9 FE              JSR PCONST
  802   FB63  30 E1                 BMI INRET         IF CAN'T ALLOCATE THE SIGN-ON
  803   FB65  A9 17                 LDA #QOUT+NEXT    QUEUE IT UP TO BE SENT
  804   FB67  A4 0E                 LDY CURBIJ
  805   FB69  20 43 FF              JSR ENQ
  806   FB6C  E6 4E                 INC CONECT        NOW CONNECTED
  807   FB6E  A5 4C                 LDA XADDR         ARE WE CONNECTING TO A SERIAL TERMINAL?
  808   FB70  29 B0                 AND #%10110000
  809   FB72  D0 02                 BNE CONYES        NO, SO CONTINUE
  810   FB74  E6 45                 INC TERMF         YES, SO SET FLAG
  811   FB76  A0 0D       CONYES    LDY #CRLF         PRINT CRLF
  812   FB78  20 63 FF              JSR PUTSTR
  813   FB7B  A0 42                 LDY #LINKA        TYPE "LINK ACCEPTED"
  814   FB7D  20 63 FF              JSR PUTSTR
  815   FB80  60          YNRET     RTS
```

```
CARD # LOC   CODE       CARD  10        20        30        40        50        60        70
  817 FB81             ;
  818 FB81             ; OUTDEV IS CALLED TO HANDLE A MESSAGE FROM THE NETWORK. IT WILL
  819 FB81             ; EITHER TRANSMIT THE MESSAGE TO THE USER DEVICE OR HANDLE THE
  820 FB81             ; MESSAGE ITSELF, DEPENDING ON THE MESSAGE TYPE.
  821 FB81             ;
  822 FB81             ; OUTDEV MAKES NO ASSUMPTION CONCERNING THE TYPE OF OPERATION UNDERWAY (I.E.
  823 FB81             ; DMA OR TTY), IT SIMPLY SENDS DATA INTO THE DEVICE AS IT IS RECEIVED.
  824 FB81             ;
  825 FB81  A5 10      OUTDEV   LDA OUTSET       SET UP FOR OUTPUT?
  826 FB83  D0 27                BNE PUTCH       YES, SO SEE IF THE NEXT CHARACTER CAN BE SENT.
  827 FB85  A6 56                LDX QIN         ANYTHING ON THE QUEUE?
  828 FB87  30 F7                BMI YYRET       NO, SO RETURN
  829 FB89  B4 7F                LDY LQPTR,X     SET UP THE PTRS
  830 FB8B  84 06                STY OUTPTR
  831 FB8D  B4 95                LDY HIPTR,X
  832 FB8F  84 07                STY OUTPTR+1
  833 FB91  A0 05                LDY #05         GET THE MSG TYPE
  834 FB93  B1 06                LDA (OUTPTR),Y
  835 FB95  C9 02                CMP #DATMSG+TTY A DATA MESSAGE?
  836 FB97  D0 55                BNE TRYSOF      NO, SO TRY FOR A SIGN-OFF MSG
  837 FB99  A4 4E                LDY CONECT      ARE WE CONNECTED?
  838 FB9B  C0 01                CPY #01
  839 FB9D  F0 03                BEQ PDATA       YES, SO PRINT THE DATA
  840 FB9F  4C 70 FC             JMP OUTFRE      NO, SO IGNORE THE PACKET
  841 FBA2             ;
  842 FBA2             ; PDATA PRINTS THE DATA IN A MSG TO THE DEVICE
  843 FBA2             ;
  844 FBA2  E6 10      PDATA    INC OUTSET      SET UP FOR OUTPUT
  845 FBA4  A0 07                LDY #07
  846 FBA6  84 08                STY OUTBC       FIRST CHAR OFFSET = 1
  847 FBA8  B1 06                LDA (OUTPTR),Y  GET THE PACKET LENGTH
  848 FBAA  85 09                STA OUTPL
  849 FBAC             ;
  850 FBAC  A4 08      PUTCH    LDY OUTBC       GET THE NEXT CHAR OFFSET
  851 FBAE  C4 09                CPY OUTPL       DONE YET?
  852 FBB0  D0 05                BNE TRYDS       NO, SO CHECK THE DEVICE STATUS
  853 FBB2  C6 10                DEC OUTSET      ALL DONE WITH THE PACKET
  854 FBB4  4C 70 FC             JMP OUTFRE
  855 FBB7             ;
  856 FBB7             ; TRYDS OUTPUTS WORDS TO THE TERMINAL TWO BYTES PER WORD.  AN ODD BYTE COUNT
  857 FBB7             ; WILL RESULT IN A PADDING BYTE BEING SUPPLIED.
  858 FBB7             ;
  859 FBB7             ; HANDSHAKING OCCURS AUTOMATICALLY WHEN DATA IS STORED IN PORT 2A.  THIS IS
  860 FBB7             ; AGAIN A HARDWARE ACTION OF THE 6522 CHIP WHICH IS EQUIVALANT TO THE G7
  861 FBB7             ; IOCTL (HARRIS 0DH/CDH).  COMMAND DATA WORDS ARE NOT PASSED TO THE G7
  862 FBB7             ;
  863 FBB7  C8         TRYDS    INY             GET THE OFFSET OF THE CHAR TO SEND
  864 FBB8  A5 45                LDA TERMF       ARE WE RECEIVING DATA FROM A SERIAL TERMINAL?
  865 FBBA  F0 04                BEQ CONDS       NO, SO CONTINUE OUTPUT
  866 FBBC  A2 00                LDX #00         YES, SO MUST SEND IT IN ONE BYTE AT  A TIME
  867 FBBE  F0 0A                BEQ CONOUT      ALWAYS BRANCH FOR LOW-ORDER BYTE
  868 FBC0  B1 06      CONDS    LDA (OUTPTR),Y  GET THE CHAR TO SEND
  869 FBC2  49 FF                EOR #$FF        TAKE THE 1'S COMPLEMENT OF THE DATA
  870 FBC4  AA                   TAX             PUT THE CHARACTER IN 'X' REG
  871 FBC5  C4 09                CPY OUTPL       DONE YET?
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  872   FBC7  F0 21              BEQ NULL          YES, SO PUT OUT A NULL CHARACTER TO FILL WORD
  873   FBC9  C8                 INY               NO, SO GET SECOND BYTE OF WORD
  874   FBCA  B1 06       COVOUT LDA (OUTPTR),Y
  875   FBCC  49 FF             EOR #$FF          TAKE THE 1'S COMPLEMENT OF THE DATA
  876   FBCE  20 D7 FB    NULRET JSR SNDOUT
  877   FBD1  F0 16              BEQ TRYLAT        INTERFACE DEADLOCK, SO TRY LATER.
  878   FBD3  84 09              STY OUTBC         SAVE BYTE COUNT
  879   FBD5  D0 D5              BNE PUTCH         GET THE NEXT WORD (ALWAYS BRANCH)
  880   FBD7                    ;
  881   FBD7                    ; SNDOUT IS USED TO SEND WORDS INTO THE GRAPHIC 7.. IN THE OUTDEV ROUTINE
  882   FBD7                    ; THE NORMAL CODE FLOW ENDS HERE, HOWEVER, IT IS USED AS A SUBROUTINE
  883   FBD7                    ; BY THE PUTSTR ROUTINE AND THE TRYSON ROUTINE.
  884   FBD7                    ;
  885   FBD7  8E 20 10    SNDOUT STX PORT2B        LOAD B WITH HIGH ORDER BYTE (NO HANDSHAKE)
  886   FBDA  8D 21 10           STA PORT2A        LOAD THE LOW ORDER BYTE (HANDSHAKE)
  887   FBDD                    ;
  888   FBDD                    ; OUTLOP WAITS FOR THE G7 TO SIGNAL IT HAS READ THE LAST WORD SENT.
  889   FBDD                    ; IT WAITS FOR ABOUT 550 USECS FOR THE RETURN HANDSHAKE.  IF THE HANDSHAKE
  890   FBDD                    ; FAILS TO COME, IT ASSUMES THERE IS AN INTERFACE DEADLOCK AND RETURNS TO
  891   FBDD                    ; THE MAIN LOOP TO ALLOW IT TO CLEAR.
  892   FBDD                    ;
  893   FBDD  A2 50              LDX #$50          SET UP THE TRY COUNTER
  894   FBDF  CA          OUTLOP DEX               COUNT THE NUMBER OF TRIES
  895   FBE0  F0 07              BEQ TRYLAT        IF TOO MANY TRIES COME BACK LATER
  896   FBE2  A9 02              LDA #DWR          TEST FOR HANDSHAKING FROM TERMINAL
  897   FBE4  2C 20 10           BIT P2IFR
  898   FBE7  F0 F6              BEQ OUTLOP        LOOP UNTIL TERMINAL ACKNOWLEDGES WORD
  899   FBE9  60           TRYLAT RTS
  900   FBEA                    ;
  901   FBEA                    ; THE NEXT TWO LINES OF CODE ADD THE NULL BYTE DURING OUTPUT.
  902   FBEA                    ;
  903   FBEA  A9 00        NULL   LDA #$00          LOAD SECOND BYTE WITH NULL CHARACTER
  904   FBEC  F0 E0              BEQ NULRET        ALWAYS BRANCH
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
  906   FBEE              ;
  907   FBEE              ; THE FOLLOWING SECTIONS CHECK FOR SPECIAL MESSAGES
  908   FBEE              ;
  909   FBEE  C9 DE       TRYSOF  CMP #SOFMSG     WAS IT A SIGN-OFF MESSAGE?
  910   FBF0  D0 32               BNE TRYSAC      NO, SO TRY FOR A SIGN-ON ACK
  911   FBF2  A5 4E               LDA CONECT      CONNECTED?
  912   FBF4  C9 01               CMP #01
  913   FBF6  D0 06               BNE CONFRE      NO, SO IGNORE THE MSG
  914   FBF8  A5 4C               LDA XADDR        IS THE PACKET FROM THE BIU TO WHICH CONNECTED?
  915   FBFA  A0 02               LDY #2
  916   FBFC  D1 06               CMP (OUTPTR),Y
  917   FBFE  D0 70       CONFRE  BNE OUTFRE       NO, SO IGNORE THE SIGN-OFF
  918   FC00  C9 21               CMP #HARBAS     IS THIS A HARRIS COMPUTER SIGN-OFF?
  919   FC02  90 09               BCC SNDLT       NO, SO CONTINUE
  920   FC04  C9 29               CMP #HARBAS+TRMCNT
  921   FC06  B0 05               BCS SNDLT       NO SO CONTINUE
  922   FC08  85 4D               STA HARGO       YES, SO MUST HAVE BEEN A POWER INTERRUPT
  923   FC0A  4C 00 F8            JMP RESET       TRY TO SIGN-ON AGAIN
  924   FC0D              ;
  925   FC0D  A0 10       SNDLT   LDY #LINKT      PRINT "LINK TERMINATED" MESSAGE
  926   FC0F  20 63 FF            JSR PUTSTR
  927   FC12  A0 00               LDY ##HSYS      ASK WHICH SYSTEM?
  928   FC14  20 63 FF            JSR PUTSTR
  929   FC17  A0 00               LDY #00         NOW WAITING FOR THE REPLY
  930   FC19  84 4C               STY XADDR
  931   FC1B  84 05               STY INBC
  932   FC1D  84 45               STY TERMF        CLEAR THE SERIAL TERMINAL FLAG
  933   FC1F  88                  DEY
  934   FC20  84 4E               STY CONECT
  935   FC22  D0 59               BNE OUTFRE      ALL DONE WITH THE MSG (ALWAYS BRANCH)
  936   FC24              ;
  937   FC24  C9 DF       TRYSAC  CMP #SACMSG     WAS IT A SIGN-ON ACK MSG?
  938   FC26  D0 24               BNE TRYSON      NO, SO TRY SIGN-ON REQUEST
  939   FC28  A4 17               LDY WAIT        WERE WE WAITING FOR THE ACK?
  940   FC2A  F0 0E               BEQ PDATAJ      NO, SO SEE IF IT HAS ANY DATA IN IT TO PRINT
  941   FC2C  C6 17               DEC WAIT        YES, SO NO LONGER WAITING
  942   FC2E  84 4E               STY CONECT      NOW CONNECTED
  943   FC30  88                  DEY             RESET SIGN-ON COUNTER
  944   FC31  84 44               STY SONCNT
  945   FC33  A0 07               LDY #07
  946   FC35  98                  TYA
  947   FC36  D1 06               CMP (OUTPTR),Y  WAS THE PACKET LENGTH 7?
  948   FC38  F0 03               BEQ PLINKA      NO, SO IT HAS DATA TO PRINT
  949   FC3A  4C A2 FB    PDATAJ  JMP PDATA
  950   FC3D  A0 42       PLINKA  LDY #LINKA      YES, SO PRINT "LINK ACCEPTED"
  951   FC3F  20 63 FF            JSR PUTSTR
  952   FC42  A5 4C               LDA XADDR       ARE WE CONNECTED TO A SERIAL TERMINAL?
  953   FC44  29 B0               AND #%10110000
  954   FC46  D0 35               BNE OUTFRE      NO, SO ALL DONE WITH THE PACKET (ALWAYS BRANCH)
  955   FC48  E6 45               INC TERMF       YES, SO SET SERIAL TERMINAL FLAG
  956   FC4A  D0 31               BNE OUTFRE      ALWAYS BRANCH
  957   FC4C              ;
  958   FC4C  C9 E0       TRYSON  CMP #SONMSG     SIGN-ON REQUEST?
  959   FC4E  90 2D               BCC OUTFRE      NO, SO IGNORE THE PACKET
  960   FC50  A4 4E               LDY CONECT      ARE WE CONNECTED?
```

```
CARD # LOC  CODE       CARD  10        20        30        40        50        60        70
  961 FC52 C0 01             CPY #01
  962 FC54 F0 27             BEQ OUTFRE      YES, SO IGNORE THE PACKET
  963 FC56 A4 17             LDY WAIT        ARE WE WAITING FOR A SIGN-ON ACK?
  964 FC58 D0 23             BNE OUTFRE      YES, SO IGNORE THE PACKET
  965 FC5A A0 1E             LDY #LINKR1     PRINT THE FIRST HALF OF 'LINK REQUEST' MESSAGE
  966 FC5C 20 63 FF          JSR PUTSTR
  967 FC5F A0 02             LDY #02         GET THE ADDR OF THE REQUESTING BIU
  968 FC61 B1 06             LDA (OUTPTR),Y  HAVE TO LOAD IT INTO A
  969 FC63 85 4C             STA XADDR       AND IN XADDR
  970 FC65 49 FF             EOR #$FF        TAKE THE 1'S COMPLEMENT OF THE DATA
  971 FC67 AA                TAX             PREPARE TO SEND IN WORD VIA SNDOUT
  972 FC68 A9 FF             LDA #$FF
  973 FC6A 20 07 FB          JSR SNDOUT
  974 FC6D A0 33             LDY #LINKR2
  975 FC6F 20 63 FF          JSR PUTSTR      PRINT THE SECOND HALF OF THE REQUEST
  976 FC72 A9 00             LDA #00
  977 FC74 85 4E             STA CONECT      NOW WAITING FOR THE Y OR N ANSWER
  978 FC76 A9 40             LDA #640        WAIT 10 SECONDS FOR A RESPONSE TO SIGN-ON
  979 FC78 A2 22             LDX #IYACK          TO PREVENT DEADLOCK
  980 FC7A 20 1A FF          JSR STIMER
  981 FC7D                ;
  982 FC7D A2 16     .OUTFRE LDX #WIN-NEXT   FREE UP THE PACKET
  983 FC7F 20 51 FF          JSR DN
  984 FC82 60                RTS             AND DONE
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
   986  FC83              ;
   987  FC83              ; TIMOUT POLLS THE TIMER TO SEE IF IT HAS TIMED OUT. IF IT HAS, IT
   988  FC83              ; DECREMENTS TICK AND POSSIBLY DECREMENTS TIMER. IF TICK REACHES 0
   989  FC83              ; (ONCE EACH SECOND) TOD IS INCREMENTED.
   990  FC83              ;
   991  FC83  2C 1D 10    TIMOUT  BIT P1IFR       HAS THE TIMER TIMED OUT?
   992  FC86  50 15               BVC TIMRET      NO
   993  FC88  AD 14 10            LDA TIMRL       YES, SO CLEAR THE FLAG
   994  FC8B  C6 4F               DEC TICK        COUNT DOWN UNTIL 1/4 SECOND HIT
   995  FC8D  D0 0E               BNE TIMRET
   996  FC8F  A9 19               LDA #25         RESET TICK TO COUNT NEXT 1/4 SECOND
   997  FC91  85 4F               STA TICK
   998  FC93  E6 19               INC TOD         INCREMENT TIME OF DAY
   999  FC95  D0 06               BNE TIMRET      IF NO CARRY, ALL DONE
  1000  FC97  E6 1A               INC TOD+1
  1001  FC99  D0 02               BNE TIMRET
  1002  FC9B  E6 1B               INC TOD+2
  1003  FC9D  60          TIMRET  RTS
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
 1005   FC9E              ;
 1006   FC9E              ; CKTOUT CHECKS FOR ANY TIMEOUTS AND TAKES APPROPRIATE ACTIONS
 1007   FC9E              ; IF ANY ARE DETECTED. FIRST, CHECK FOR TIMEOUT FOR SIGN-ON ACK.
 1008   FC9E              ;
 1009   FC9E  A5 17       CKTOUT  LDA WAIT          ARE WE WAITING FOR A SIGN-ON ACK?
 1010   FCA0  F0 35               BEQ CKYOUT        NO, SO CHECK FOR YES-NO RESPONSE TIMEOUT
 1011   FCA2  A2 1F               LDX #TSACK
 1012   FCA4  20 03 FD            JSR CTIME
 1013   FCA7  90 2E               BCC CKYOUT        CARRY CLEAR INDICATES NOT TIMED OUT
 1014   FCA9              ;
 1015   FCA9              ; THE FOLLOWING CODE WILL CAUSE A NUMBER OF SIGN-ON REQUESTS BASED ON
 1016   FCA9              ; TRMCNT TO BE SENT TO FIND AN UNUSED HARRIS BIU.
 1017   FCA9              ;
 1018   FCA9  A6 4C               LDX XADDR         START WITH THE BASE HARRIS ADDRESS FOR THIS BIU
 1019   FCAB  E0 21               CPX #HARBAS       IS IT WITHIN THE ALLOWED RANGE
 1020   FCAD  90 1E               BCC NOSYS         NOT TRYING TO SIGN-ON TO HARRIS BIU
 1021   FCAF  E0 29               CPX #HARBAS+TRMCNT
 1022   FCB1  B0 12               BCS RSXADR        LOOP BACK TO THE BASE HARRIS ADDRESS
 1023   FCB3  E6 44               INC SONCNT        COUNT THE NUMBER OF TIMES SIGN-ON TRIED
 1024   FCB5  A4 44               LDY SONCNT
 1025   FCB7  C0 09               CPY #TRMCNT       HAVE WE TRIED ALL OF THE HARRIS BIUS?
 1026   FCB9  F0 12               BEQ NOSYS         YES, SO NONE AVAILABLE
 1027   FCBB  E8                  INX               NO, SO TRY NEXT ONE
 1028   FCBC  A9 02       CONSON  LDA #2            USE A PORTION OF INDEV ROUTINE
 1029   FCBE  C6 17               DEC WAIT          RESET THE WAIT FLAG
 1030   FCC0  20 2C FB            JSR SYSTMX        ... BEGINNING AT SYSTMX
 1031   FCC3  D0 12               BNE CKYOUT        CONTINUE WITH NEXT TIMER CHECK
 1032   FCC5              ;
 1033   FCC5  A5 44       RSXADR  LDA SONCNT         CHECK TO SEE IF THIS IS A HARRIS SIGN-ON
 1034   FCC7  F0 04               BEQ NOSYS          NO, SO SEND SYSTEM NOT AVAILABLE.
 1035   FCC9  A2 21               LDX #HARBAS       START WITH BASE ADDRESS NOW
 1036   FCCB  D0 EF               BNE CONSON
 1037   FCCD              ;
 1038   FCCD  A0 4F       NOSYS   LDY #NOVAIL       PRINT "SYSTEM NOT AVAILABLE"
 1039   FCCF  20 63 FF            JSR PUTSTR
 1040   FCD2  85 44               STA SONCNT         RESET THE SIGN-ON COUNTER
 1041   FCD4  4C F5 FA            JMP ASKAGN
 1042   FCD7              ;
 1043   FCD7              ; SEE IF WE WAITED LONG ENOUGH TO RESPOND TO A SIGN-ON REQUEST.
 1044   FCD7              ;
 1045   FCD7  A5 4E       CKYOUT  LDA CONECT        WAITING FOR A SIGN-ON Y-N?
 1046   FCD9  D0 13               BNE CKETTY        NO, SO SEE IF TIME TO RESET TTY FLAG.
 1047   FCDB  A2 22               LDX #TYACK
 1048   FCDD  20 03 FD            JSR CTIME
 1049   FCE0  90 0C               BCC CKETTY
 1050   FCE2  C6 4E               DEC CONECT        NO LONGER CONNECTED
 1051   FCE4  A0 4F               LDY #NOVAIL       PRINT "SYSTEM NOT AVAILABLE"
 1052   FCE6  20 63 FF            JSR PUTSTR
 1053   FCE9  A0 00               LDY #WHSYS        ASK WHICH SYSTEM AGAIN
 1054   FCEB  20 63 FF            JSR PUTSTR
 1055   FCEE              ;
 1056   FCEE              ; SEE IF THE TTY TIMER FLAG SHOULD BE RESET.
 1057   FCEE              ;
 1058   FCEE  A6 43       CKETTY  LDX TTYTM         HAS THE TIMER FLAG BEEN SET?
 1059   FCF0  F0 09               BEQ CKSTAT        NO, SO SEE IF IT IS TIME TO SEND A STATUS MESSAGE.
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
 1060   FCF2  A2 25            LDX #TETTY       YES, SO SEE IF TIMER HAS EXPIRED
 1061   FCF4  20 03 FD         JSR CTIME
 1062   FCF7  90 02            BCC CKSTAT
 1063   FCF9  C6 43            DEC TTYT4
 1064   FCFB                ;
 1065   FCFB                ; SEE IF IT IS TIME TO SEND A STATUS MESSAGE.
 1066   FCFB                ;
 1067   FCFB  A2 1C       CKSTAT   LDX #TSTAT
 1068   FCFD  20 03 FD             JSR CTIME
 1069   FD00  B0 16                BCS SNDSTA
 1070   FD02  60          CKCRET   RTS
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
 1072   FD03             ;
 1073   FD03             ; CTIME IS USED TO COMPARE THE CURRENT TIME OF DAY TO THE VALUE
 1074   FD03             ; STORED IN THE VARIABLE X,X+1 AND X+2.(3 BYTES REQUIRED).  IF THE
 1075   FD03             ; TOD IS GREATER THAN OR EQUAL TO THE VARIABLE CARRY WILL BE SET
 1076   FD03             ; UPON RETURN; OTHERWISE CARRY IS CLEARED.
 1077   FD03             ;
 1078   FD03  A5 18      CTIME   LDA TOD+2
 1079   FD05  D5 02              CMP 2,X
 1080   FD07  90 0E              BCC CTRET
 1081   FD09  D0 0C              BNE CTRET       TOD LESS THAN X
 1082   FD0B  A5 1A              LDA TOD+1       GET SECOND BYTE
 1083   FD0D  D5 01              CMP 1,X
 1084   FD0F  90 06              BCC CTRET
 1085   FD11  D0 04              BNE CTRET       TOD LESS THAN X
 1086   FD13  A5 19              LDA TOD
 1087   FD15  D5 00              CMP 0,X
 1088   FD17  60         CTRET   RTS
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
 1090   FD18              ;
 1091   FD18              ; SNDSTA IS USED TO SEND A STATUS MESSAGE
 1092   FD18              ;
 1093   FD18  A9 D8       SNDSTA  LDA #STAMSG     SEND A STATUS MSG
 1094   FD1A  20 E9 FE            JSR PCONST      TRY TO GET A BUFFER
 1095   FD1D  30 E3               BMI CKCRET      NOT ALLOCATED, SO TRY LATER
 1096   FD1F  A0 00               LDY #00         TO ADDRESS $00
 1097   FD21  98                  TYA
 1098   FD22  91 0A               STA (BIUPTR),Y
 1099   FD24  A9 29               LDA #41         PACKET LENGTH OF 42
 1100   FD26  A0 07               LDY #07
 1101   FD28  91 0A               STA (BIUPTR),Y
 1102   FD2A              ;
 1103   FD2A  C8                  INY             POINT TO THE FIRST BYTE OF STATUS PART
 1104   FD2B  A2 00               LDX #00         MOVE IN THE 16 BYTES OF STATUS INFORMATION
 1105   FD2D  B5 28       SLOOP1  LDA NMXMT,X     GET A BYTE
 1106   FD2F  91 0A               STA (BIUPTR),Y  AND SAVE IT IN THE MESSAGE
 1107   FD31  E8                  INX
 1108   FD32  C8                  INY
 1109   FD33  E0 10               CPX #16         SIXTEEN BYTES MOVED YET?
 1110   FD35  90 F6               BCC SLOOP1      BRANCH IF NO
 1111   FD37              ;
 1112   FD37  A9 00               LDA #00         GET THE # OF MSGS WAITING TO GO OUT
 1113   FD39  A6 67               LDX QOUT
 1114   FD3B  30 0C               BMI S2OT2       IF NONE
 1115   FD3D  18          SLOOP2  CLC             INCR # WAITING
 1116   FD3E  69 01               ADC #01
 1117   FD40  48                  PHA             SAVE A
 1118   FD41  B5 50               LDA NEXT,X      GET THE PTR TO THE NEXT MSG
 1119   FD43  AA                  TAX
 1120   FD44  68                  PLA             RESTORE A
 1121   FD45  E0 FF               CPX #$FF        IS THE NEXT PTR NULL?
 1122   FD47  D0 F4               BNE SLOOP2      NO, SO GET THE NEXT ONE
 1123   FD49  91 0A       S8OT2   STA (BIUPTR),Y  SAVE THE # OF WAITING MSGS
 1124   FD4B              ;
 1125   FD4B  A9 00               LDA #00         GET THE # OF MSGS WAITING TO GO IN
 1126   FD4D  A6 66               LDX QIN
 1127   FD4F  30 0C               BMI S8OT3       IF NONE
 1128   FD51  18          SLOOP3  CLC
 1129   FD52  69 01               ADC #01
 1130   FD54  48                  PHA
 1131   FD55  B5 50               LDA NEXT,X
 1132   FD57  AA                  TAX
 1133   FD58  68                  PLA
 1134   FD59  E0 FF               CPX #$FF
 1135   FD5B  D0 F4               BNE SLOOP3
 1136   FD5D  C8          S8OT3   INY             SAVE THE # OF WAITING MSGS
 1137   FD5E  91 0A               STA (BIUPTR),Y
 1138   FD60              ;
 1139   FD60  C8                  INY
 1140   FD61  AD 00 0C            LDA NUARTS      NETWORK UART STATUS
 1141   FD64  91 0A               STA (BIUPTR),Y
 1142   FD66  C8                  INY
 1143   FD67  AD 00 14            LDA DUARTS      DEVICE UART STATUS
 1144   FD6A  91 0A               STA (BIUPTR),Y
```

```
CARD # LOC    CODE        CARD  10        20        30        40        50        60        70
 1145  FD6C  C8                  INY
 1146  FD6D  AD 1D 10            LDA P1IFR       PARALLEL PORT INTERRUPT FLAG REGISTER
 1147  FD70  91 0A               STA (BIUPTR),Y
 1148  FD72  C8                  INY
 1149  FD73  A5 4C               LDA XADDR       THE CURRENT XMIT ADDRESS
 1150  FD75  91 0A               STA (BIUPTR),Y
 1151  FD77              ;
 1152  FD77  C8                  INY             POINT TO THE FIRST BYTE OF THE DESCRIPTOR
 1153  FD78              ;
 1154  FD78              ; REG 'X' EQUALS -1 FROM LOOP ABOVE.
 1155  FD78              ;
 1156  FD78  E8                  INX             MOVE IN 12 BYTES OF BIU FUNCTION CODE
 1157  FD79  BD EA FF    SLOOP4  LDA BIUFNC,X    GET A BYTE
 1158  FD7C  91 0A               STA (BIUPTR),Y  AND SAVE IT IN THE MESSAGE
 1159  FD7E  E8                  INX             UPDATE THE POINTERS AND LOOP COUNTER
 1160  FD7F  C8                  INY
 1161  FD80  E0 0C               CPX #12         TWELVE BYTES MOVED YET?
 1162  FD82  90 F5               BCC SLOOP4      BRANCH IF NO
 1163  FD84              ;
 1164  FD84  A4 17               LDA #OOUT-NEXT  SEND THE STATUS MSG
 1165  FD86  A4 0E               LDY CURBIU
 1166  FD88  20 43 FF            JSR ENQ
 1167  FD8B  A9 F0               LDA #240        THE NEXT STATUS MSG GOES OUT IN 1 MINUTE
 1168  FD8D  A2 1C               LDX #TSTAT
 1169  FD8F  20 1A FF            JSR STIMER
 1170  FD92              ;
 1171  FD92  A9 00               LDA #00         CLEAR THE STATUS COUNTERS AND RETURN
 1172  FD94  A2 0F               LDX #15
 1173  FD96  95 28       SLOOP5  STA NMXMT,X
 1174  FD98  CA                  DEX
 1175  FD99  10 FB               BPL SLOOP5
 1176  FD9B              ;
 1177  FD9B  60                  RTS
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
 1179   FD9C              ;
 1180   FD9C              ; INTBUF SETS UP A BUFFER FOR NETWORK INPUT IF IT IS NECESSARY.
 1181   FD9C              ;
 1182   FD9C  08          INTBUF  PHP               PROTECTS THE INTERRUPT MASKING.
 1183   FD9D  78                  SEI               DON'T WANT INTERRUPTS DURING ALLOC
 1184   FD9E  A5 11               LDA INTSET
 1185   FDA0  D0 11               BNE INTRES        ALREADY HAVE A BUFFER, SO RETURN
 1186   FDA2  20 39 FF            JSR ALLOC         ALLOCATE A BUFFER
 1187   FDA5  30 0C               BMI INTRES        NO BUFFERS AVAILABLE, SO TRY AGAIN LATER
 1188   FDA7  B5 7F               LDA LOPTR,X       SET UP THE PTR TO THE BUFFER
 1189   FDA9  85 00               STA INTPTR
 1190   FDAB  B5 95               LDA HIPTR,X
 1191   FDAD  85 01               STA INTPTR+1
 1192   FDAF  86 0C               STX CURNET        SAVE THE BUFFER NUMBER
 1193   FDB1  E6 11               INC INTSET        NOW ALL SET UP
 1194   FDB3  28          INTRES  PLP               RESTORE OLD INTERRUPT STATUS AND RETURN
 1195   FDB4  60                  RTS
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
 1197   FDB5              ;
 1198   FDB5              ; IRQ RECEIVES CONTROL WHENEVER THERE IS A NETWORK INTERRUPT
 1199   FDB5              ; THIS CAN BE GENERATED BY THE RECEIVE DATA REGISTER OF THE NETWORK UART
 1200   FDB5              ; BECOMING FULL, THE DATA CARRIER DETECT SIGNAL GOING HIGH, OR A DATA
 1201   FDB5              ; OVERRUN CONDITION.
 1202   FDB5              ;
 1203   FDB5  48          IRQ       PHA             PUSH ACC, Y AND X ONTO STACK
 1204   FDB6  98                    TYA
 1205   FDB7  48                    PHA
 1206   FDB8  8A                    TXA
 1207   FDB9  48                    PHA
 1208   FDBA              ;
 1209   FDBA              ; THIS CODE IS THE NETWORK UART INTERRUPT HANDLER. IT IS USED TO EITHER
 1210   FDBA              ; CHECK XMT VALIDITY, OR TO READ IN A MSG FROM ANOTHER BIU.
 1211   FDBA              ;
 1212   FDBA  AC 00 0C              LDY NUARTS      GET THE STATUS (CLEARS THE DATA OVERRUN IF NECESSARY)
 1213   FDBD  AD 01 0C              LDA NUARTD      AND THE DATA
 1214   FDC0  84 02                 STY INTBC       SAVE THE PARITY ERROR FLAG
 1215   FDC2  A6 3F                 LDX INTX        INTX IS INITIALLY ZERO
 1216   FDC4  A4 14                 LDY XMIT        IS THIS MY TRANSMISSION?
 1217   FDC6  F0 3F                 BEQ RXDATA      IF NOT, GO TO N4
 1218   FDC8  95 15                 STA TOKEN,X     SAVE RECEIVED DATA IN TOKEN
 1219   FDCA  E8                    INX
 1220   FDCB  E0 02                 CPX #02         HAVE 2 CHARACTERS BEEN RECEIVED?
 1221   FDCD  F0 03                 BEQ TESTTX      YES, SO TEST THEM
 1222   FDCF  4C CC FE              JMP NIRET       NO, SO HAVE TO WAIT, BUT KEEP DECREMENTING INTX
 1223   FDD2              ;
 1224   FDD2  A0 01       TESTTX    LDY #01         TEST RCV'D VS. TRANSMITTED DATA
 1225   FDD4  B9 15 00    TEST2     LDA TOKEN,Y
 1226   FDD7  D1 06                 CMP (OUTPTR),Y  OUTPTR POINTS TO ANY PACKET BEING XMITTED ON NET
 1227   FDD9  D0 0B                 BNE COLIDE
 1228   FDDB  88                    DEY
 1229   FDDC  10 F6                 BPL TEST2
 1230   FDDE  A9 18                 LDA #%00011000  TEST OK, TURN OFF NET RCVR
 1231   FDE0  8D 00 0C              STA NUARTS
 1232   FDE3  4C CA FE              JMP NICLRX
 1233   FDE6              ;
 1234   FDE6  A9 58       COLIDE    LDA #%01011000  TEST FAILED, TURN OFF XMIT AND RCV
 1235   FDE8  8D 00 0C              STA NUARTS
 1236   FDEB  A5 12                 LDA RANDU       CHOOSE RANDOM WAIT
 1237   FDED  0A                    ASL A           TO GET NEXT RANDOM NUMBER, ...
 1238   FDEE  0A                    ASL A            MULTIPLY BY 13 AND ADD 1
 1239   FDEF  85 02                 STA INTBC       X4
 1240   FDF1  0A                    ASL A           X8
 1241   FDF2  18                    CLC
 1242   FDF3  65 02                 ADC INTBC       X12
 1243   FDF5  18                    CLC
 1244   FDF6  65 12                 ADC RANDU       X13
 1245   FDF8  18                    CLC
 1246   FDF9  69 01                 ADC #01         +1
 1247   FDFB  85 12                 STA RANDU       STORE NEW RANDOM NUMBER AT SEED AND COUNT
 1248   FDFD  85 13                 STA RNDCNT
 1249   FDFF  A2 2C                 LDX #NMCOL      INCREMENT # OF COLLISIONS
 1250   FE01  20 2C FF              JSR SFINC
 1251   FE04  4C CA FE              JMP NICLRX
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
 1252   FE07           ;
 1253   FE07  CD 40 10 RXDATA   CMP PORT3B      IS THE PACKET FOR US?
 1254   FE0A  F0 08            BEQ FORUS       YES
 1255   FE0C           ;
 1256   FE0C  A9 58    SKIPIT  LDA #%01011000  CAN'T RECEIVE, SO DISABLE RECEIVER
 1257   FE0E  8D 00 0C         STA NUARTS
 1258   FE11  4C CA FE         JMP NICLRX      AND RETURN
 1259   FE14           ;
 1260   FE14  24 02    FORUS   BIT INTBC       WAS THERE A PARITY ERROR IN THE ADDRESS?
 1261   FE16  70 F4            BVS SKIPIT      YES, SO IGNORE THE PACKET
 1262   FE18  A4 11            LDY INTSET      ARE WE ALL SET UP FOR INPUT?
 1263   FE1A  D0 10            BNE GETIT       YES
 1264   FE1C  A2 34            LDX #NMLOST     INCR THE # OF MSGS LOST BECAUSE OF NO BUFFER
 1265   FE1E  20 2C FF         JSR SFINC
 1266   FE21  A9 58            LDA #%01011000  TURN OFF THE RECEIVER
 1267   FE23  8D 00 0C         STA NUARTS
 1268   FE26           ;
 1269   FE26           ;  SET UP TO SEND AN $FF ACK ...
 1270   FE26           ;
 1271   FE26  A9 FF            LDA #$FF
 1272   FE28  85 46            STA ACKBYT      TO SIGNAL THAT WE ARE FULL AND THAT HE
 1273   FE2A  D0 52            BNE SNDACK      SEND THE STOP-XMIT FLAG (ALWAYS BRANCH)
 1274   FE2C           ;
 1275   FE2C  A0 00    GETIT   LDY #00         STORE BYTE 0 IN BUFFER
 1276   FE2E  91 00            STA (INTPTR),Y
 1277   FE30  85 46            STA ACKBYT       SET UP TO SEND A GOOD ACK PACKET
 1278   FE32  85 38            STA PARITY      START THE VERTICAL PARITY CALCULATION
 1279   FE34  C8       GETBYT  INY
 1280   FE35  AD 00 0C WAITBY  LDA NUARTS      IS ANOTHER BYTE READY
 1281   FE38  29 05            AND #%00000101  IS RECEIVE KEY ON BUT WORD NOT IN?
 1282   FE3A  F0 F9            BEQ WAITBY      IF NOT, GO TO WAITBY
 1283   FE3C  29 04            AND #%00000100  IS RECEIVE KEY ON?
 1284   FE3E  D0 CC            BNE SKIPIT      IF NOT, TURN OFF THE RECEIVER
 1285   FE40  2C 00 0C         BIT NUARTS      WAS THERE A PARITY ERROR?
 1286   FE43  70 32            BVS PARERR      YES, SO ABORT THIS MESSAGE
 1287   FE45  AD 01 0C         LDA NUARTD      READ THE NEXT WORD
 1288   FE48  91 00            STA (INTPTR),Y
 1289   FE4A  AA               TAX             SAVE THE DATA CHAR
 1290   FE4B  45 38            EOR PARITY      ACCUMULATE THE PARITY
 1291   FE4D  85 38            STA PARITY
 1292   FE4F  C0 07            CPY #07         IS THIS PACKET BYTE COUNT?
 1293   FE51  90 E1            BCC GETBYT      IF <, KEEP ON READING IN THE HEADER
 1294   FE53  D0 02            BNE INLOOP      IF >, COMPARE TO INTBC
 1295   FE55  86 02            STX INTBC       IF =, STORE RECEIVE PACKET LENGTH
 1296   FE57  C4 02    INLOOP  CPY INTBC       DOES Y = PACKET LENGTH?
 1297   FE59  D0 D9            BNE GETBYT      NO, SO KEEP ON GETTING BYTES
 1298   FE5B           ;
 1299   FE5B  AD 00 0C CHKLOP  LDA NUARTS      WAIT FOR THE CHECKSUM
 1300   FE5E  29 05            AND #%00000101
 1301   FE60  F0 F9            BEQ CHKLOP
 1302   FE62  29 04            AND #%00000100
 1303   FE64  D0 A6            BNE SKIPIT
 1304   FE66  2C 00 0C         BIT NUARTS      BAD PARITY?
 1305   FE69  70 0C            BVS PARERR      YES
 1306   FE6B  AE 01 0C         LDX NUARTD      GET THE VERTICAL PARITY
```

```
CARD # LOC   CODE       CARD  10        20        30        40        50        60        70
 1307  FE6E  A9 58            LDA #%01011000  DISABLE THE RECEIVER
 1308  FE70  8D 00 0C         STA NUARTS
 1309  FE73  E4 38            CPX PARITY      COMPARE THE PARITIES
 1310  FE75  F0 07            BEQ SNDACK      HURRAH, THEY AGREE
 1311  FE77  A2 32     PARERR LDX #NMBCRC     KEEP COUNT OF PARITY ERRORS
 1312  FE79  20 2C FF         JSR SFINC
 1313  FE7C  D0 BE            BNE SKIPIT      DISABLE THE RECEIVER AND RETURN (ALWAYS BRANCH)
 1314  FE7E                  ;
 1315  FE7E  AD 00 0C  SNDACK LDA NUARTS      IS THE NET BUSY?
 1316  FE81  29 04            AND #%00000100
 1317  FE83  F0 F9            BEQ SNDACK      YES
 1318  FE85  A9 18            LDA #%00011000  TURN ON THE XMITTER
 1319  FE87  8D 00 0C         STA NUARTS
 1320  FE8A  A5 46            LDA ACKBYT      SEND THE ACK
 1321  FE8C  8D 01 0C         STA NUARTD
 1322  FE8F  A2 1A            LDX #26         WAIT 75 US BEFORE TURNING OFF THE XMIT KEY
 1323  FE91  CA       WAIT75  DEX
 1324  FE92  10 FD            BPL WAIT75
 1325  FE94  A2 58            LDX #%01011000  TURN OFF THE XMITTER
 1326  FE96  8E 00 0C         STX NUARTS
 1327  FE99  C9 FF            CMP #$FF        DID WE SEND THE STOP-XMIT FLAG?
 1328  FE9B  F0 2A            BEQ TXNACK      YES, GO TRY ALLOCATION & RETURN
 1329  FE9D                  ;
 1330  FE9D  A0 02            LDY #02         GET THE XMIT ADDR
 1331  FE9F  31 00            LDA (INTPTR),Y
 1332  FEA1  C5 39            CMP LASTTX      SAME AS THE PREVIOUSLY RECEIVED MESSAGE?
 1333  FEA3  D0 08            BNE MSGOK       NO, SO THE MSG IS OK
 1334  FEA5  A0 04            LDY #04         GET THE SEQUENCE #
 1335  FEA7  B1 00            LDA (INTPTR),Y
 1336  FEA9  C5 3C            CMP LASTSQ      SAME AS THE PREVIOUS MSG?
 1337  FEAB  F0 19            BEQ NICLRX      YES, SO SKIP IT
 1338  FEAD                  ;
 1339  FEAD  A0 02     MSGOK  LDY #02         SAVE THE NEW TRANSMIT ADDRESS
 1340  FEAF  B1 00            LDA (INTPTR),Y
 1341  FEB1  85 3B            STA LASTTX
 1342  FEB3  A0 04            LDY #04
 1343  FEB5  B1 00            LDA (INTPTR),Y  THE NEW SEQUENCE #,
 1344  FEB7  85 3C            STA LASTSQ
 1345  FEB9                  ;
 1346  FEB9  A9 16            LDA #QIV-NEXT   QUEUE THE MESSAGE TO BE SENT IN TO THE DEVICE
 1347  FEBB  A4 0C            LDY CURNET
 1348  FEBD  20 43 FF         JSR ENQ
 1349  FEC0  C6 11            DEC INTSET      NO LONGER SET UP FOR INPUT
 1350  FEC2  A2 30            LDX #NMGCRC     INCR # OF MESSAGES RECEIVED WITH GOOD CKSUM
 1351  FEC4  20 2C FF         JSR SFINC
 1352  FEC7                  ;
 1353  FEC7  20 9C FD  TXNACK JSR INTBUF      TRY TO GET SET UP AGAIN
 1354  FECA  A2 00     NICLRX LDX #00
 1355  FECC  86 3F     NIRET  STX INTX        SAVE OUR COPY OF X, OR CLEAR IT, AS NEEDED
 1356  FECE  A2 00            LDX #00
 1357  FED0  86 40            STX INTFLG
 1358  FED2                  ;
 1359  FED2                  ; THIS CODE IS USED TO RESTORE REGISTERS AND RETURN FROM INTERRUPTS.
 1360  FED2                  ;
 1361  FED2  68               PLA             UNSTACK AND RETURN
```

```
CARD #  LOC   CODE      CARD  10        20        30        40        50        60        70
 1362   FED3  AA              TAX
 1363   FED4  68              PLA
 1364   FED5  A8              TAY
 1365   FED6  68              PLA
 1366   FED7  40              RTI
```

```
CARD # LOC  CODE       CARD  10        20        30        40        50        60        70
 1368  FED8           ;
 1369  FED8           ; NMI OCCURS WHEN THE RECEIVE KEY TURNS OFF.(I.E. WHEN THE CARRIER ON THE
 1370  FED8           ; CABLE GOES LOW.)
 1371  FED8           ;
 1372  FED8  48       NMI       PHA               PUSH A
 1373  FED9  A9 5B               LDA #%01011011    RESET NETWORK UART
 1374  FEDB  8D 00 0C            STA NUARTS
 1375  FEDE  A9 D8               LDA #%11011000    INITIALIZE NETWORK UART
 1376  FEE0  8D 00 0C            STA NUARTS
 1377  FEE3  A9 00               LDA #00           SET XMIT FLAG TO 0
 1378  FEE5  85 14               STA XMIT
 1379  FEE7  68                  PLA
 1380  FEE8  40                  RTI
```

```
CARD #  LOC   CODE        CARD  10        20        30        40        50        60        70
 1382   FEE9                ;
 1383   FEE9                ; PCONST IS USED TO CONSTRUCT AN OUTGOING PACKET.
 1384   FEE9                ; IT EXPECTS THE MESSAGE TYPE TO BE PASSED IN A.
 1385   FEE9                ; IF IT CAN'T ALLOCATE A BUFFER, IT RETURNS WITH THE NEGATIVE BIT ON.
 1386   FEE9                ;
 1387   FEE9  78            PCONST  SEI             DON'T WANT INTERRUPTS DURING ALLOC.
 1388   FEEA  20 39 FF              JSR ALLOC       GET A BUFFER
 1389   FEED  58                    CLI
 1390   FEEE  30 29                 BMI PRET        COULDN'T GET A BUFFER, SO RETURN
 1391   FEF0  B4 7F                 LDY LOPTR,X     GET THE PTR TO THE BUFFER.
 1392   FEF2  84 0A                 STY BIUPTR
 1393   FEF4  B4 95                 LDY HIPTR,X
 1394   FEF6  84 0B                 STY BIUPTR+1
 1395   FEF8  86 0E                 STX CURBIU      SAVE THE BUFFER #
 1396   FEFA  AA                    TAX
 1397   FEFB  A9 00                 LDA #00         CLEAR THE MESSAGE HEADER
 1398   FEFD  A0 06                 LDY #06
 1399   FEFF  91 0A         CLOOP   STA (BIUPTR),Y
 1400   FF01  88                    DEY
 1401   FF02  D0 FB                 BNE CLOOP
 1402   FF04                ;
 1403   FF04  A5 4C                 LDA XADDR
 1404   FF06  91 0A                 STA (BIUPTR),Y  STORE THE DESTINATION ADDRESS (Y = 0)
 1405   FF08  A0 02                 LDY #02
 1406   FF0A  AD 40 10              LDA PORT3B
 1407   FF0D  91 0A                 STA (BIUPTR),Y  STORE THE TRANSMIT ADDRESS
 1408   FF0F  8A                    TXA
 1409   FF10  A0 05                 LDY #05
 1410   FF12  91 0A                 STA (BIUPTR),Y  STORE THE MESSAGE TYPE
 1411   FF14  A9 07                 LDA #07         DEFAULT PACKET LENGTH OF 7
 1412   FF16  A8                    TAY
 1413   FF17  91 0A                 STA (BIUPTR),Y  STORE IT IN THE PACKET
 1414   FF19  60            PRET    RTS
 1415   FF1A                ;
 1416   FF1A                ; STIMER SETS A TIMER VARIABLE TO AN INCREMENT PLUS THE CURRENT
 1417   FF1A                ; VALUE OF TOD. A IS THE INCREMENT TO ADD TO TOD AND X POINTS TO
 1418   FF1A                ; THE VARIABLE THAT HOLDS THE RESULT, LOW BYTE FIRST. A IS CHANGED.
 1419   FF1A                ;
 1420   FF1A  18            STIMER  CLC
 1421   FF1B  65 19                 ADC TOD         ADD TOD TO THE INCR IN A
 1422   FF1D  95 00                 STA 0,X         SAVE IT IN THE LOW RESULT BYTE
 1423   FF1F  A5 1A                 LDA TOD+1       ADD IN THE CARRY
 1424   FF21  69 00                 ADC #00
 1425   FF23  95 01                 STA 1,X         SAVE IT IN THE SECOND BYTE
 1426   FF25  A5 1B                 LDA TOD+2
 1427   FF27  69 00                 ADC #00
 1428   FF29  95 02                 STA 2,X         AND IN THE HIGH BYTE
 1429   FF2B  60                    RTS
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
 1431   FF2C             ;
 1432   FF2C             ; SFINC IS CALLED TO INCREMENT THE CONTENTS OF A STATUS MSG FIELD. THE
 1433   FF2C             ; FIELD WILL NOT WRAP AROUND TO 0000. X => THE FIELD (HI BYTE FIRST).
 1434   FF2C             ;
 1435   FF2C  F6 01      SFINC   INC 1,X           INCREMENT THE LOWER BYTE
 1436   FF2E  D0 08             BNE SFRET         IF NO CARRY, ALL DONE
 1437   FF30  F6 00             INC 0,X           INCREMENT THE HIGHER BYTE
 1438   FF32  D0 04             BNE SFRET         IF NOT ZERO, THEN ALL DONE
 1439   FF34  D6 01             DEC 1,X           WRAPPED AROUND, SO RETURN TO FFFF
 1440   FF36  D6 00             DEC 0,X
 1441   FF38  60        SFRET   RTS
 1442   FF39             ;
 1443   FF39             ; ALLOC IS CALLED TO ALLOCATE A FREE BUFFER AND RETURN ITS NUMBER IN
 1444   FF39             ; X. IF THERE ARE NONE LEFT, THE N BIT WILL BE ON. X AND Y ARE CHANGED.
 1445   FF39             ;
 1446   FF39  A4 3D      ALLOC   LDY STKPTR        GET OFFSET OF THE TOP OF FREE BUFFER STACK + 1
 1447   FF3B  88                 DEY               POINT TO THE NEXT FREE BUFFER NUMBER
 1448   FF3C  30 04              BMI ALRET         RETURN IF NONE AVAILABLE
 1449   FF3E  84 3D              STY STKPTR        GOT ONE, SO SAVE THE NEW TOP OF THE STACK
 1450   FF40  B6 69              LDX BUFSTK,Y      GET THE ALLOCATED BUFFER NUMBER
 1451   FF42  60        ALRET   RTS
 1452   FF43             ;
 1453   FF43             ; ENQ IS USED TO ADD A BUFFER TO THE END OF A QUEUE.  A IS THE OFFSET OF
 1454   FF43             ; THE QUEUE HEADER FROM NEXT, Y IS THE # OF THE BUFFER.  X & A ARE CHANGED.
 1455   FF43             ;
 1456   FF43  08        ENQ     PHP               TURN OFF INTERRUPTS
 1457   FF44  78                SEI
 1458   FF45  AA        ENQ1    TAX               STEP ONE FURTHER ALONG THE QUEUE
 1459   FF46  B5 50             LDA NEXT,X        IS THIS THE LAST ENTRY?
 1460   FF48  10 F8             BPL ENQ1          NO, SO KEEP ON LOOKING
 1461   FF4A  94 50             STY NEXT,X        SET LINK IN LAST ENTRY TO NEW ENTRY
 1462   FF4C  99 50 00          STA NEXT,Y        SET THE NEXT PTR IN THE NEW ENTRY TO NULL
 1463   FF4F  28                PLP
 1464   FF50  60                RTS
 1465   FF51             ;
 1466   FF51             ; DQ IS CALLED TO DEQUEUE AND FREE A BUFFER ON A QUEUE. X IS THE OFFSET
 1467   FF51             ; FROM NEXT OF ENTRY PREVIOUS TO THE ONE TO BE FREED. X,A,Y ARE CHANGED
 1468   FF51             ; THIS DQ DIFFERS FROM THE MODCOMP BIU DQ IN THAT IT LACKS THE FREE
 1469   FF51             ; ENTRY POINT
 1470   FF51             ;
 1471   FF51  08        DQ      PHP
 1472   FF52  78                SEI               TURN OFF INTERRUPTS.
 1473   FF53  B4 50             LDY NEXT,X        Y HAS THE NUMBER OF THE BUFFER TO BE FREED
 1474   FF55  B9 50 00          LDA NEXT,Y        A HAS THE NUMBER OF THE NEXT BUFFER IN THE QUEUE
 1475   FF58  95 50             STA NEXT,X        THE BUFFER IS DEQUEUED
 1476   FF5A  A6 3D             LDX STKPTR        ADD THE BUFFER TO THE FREE BUFFER STACK
 1477   FF5C  98                TYA
 1478   FF5D  95 69             STA BUFSTK,X      NOW IN THE FREE BUFFER POOL
 1479   FF5F  E6 3D             INC STKPTR        ONE MORE FREE BUFFER
 1480   FF61  28                PLP
 1481   FF62  60                RTS
```

```
CARD #  LOC   CODE       CARD  10        20        30        40        50        60        70
 1483   FF63              ;
 1484   FF63              ; PUTSTR IS USED TO SEND IN A STRING FROM THE ASCII LIST TO THE G7.
 1485   FF63              ; IT EXPECTS THE OFFSET OF THE STRING FROM THE BEGINNING OF ASCII IN REG 'Y'
 1486   FF63              ;
 1487   FF63  A2 FF       PUTSTR  LDX #$FF          INITIALIZE NULL LEADING CHARACTER
 1488   FF65  B9 93 FF            LDA ASCII,Y       GET A CHARACTER FROM THE STRING
 1489   FF68  F0 08               BEQ PUTRET        IF LAST CHARACTER MARKED BY A 00, RETURN
 1490   FF6A  49 FF               EOR #$FF           TAKE THE 1'S COMPLEMENT OF THE DATA
 1491   FF6C  20 D7 FB            JSR SNDOUT
 1492   FF6F  C8                  INY               SET UP FOR NEXT BYTE
 1493   FF70  D0 F1               BNE PUTSTR         GO GET NEXT CHARACTER
 1494   FF72  60          PUTRET  RTS
```

```
CARD # LOC  CODE        CARD 10        20        30        40        50        60        70
 1496 FF73              ;
 1497 FF73              ; CONSTANTS AND TABLES FOLLOW
 1498 FF73              ;
 1499 FF73              ; SYSTEM TABLE FOLLOWS, IT HAS EACH SYSTEM NAME, ITS DESTINATION
 1500 FF73              ; ADDRESS, OR A NULL ADDRESS IF THE ADDRESS IS SUPPLIED BY A SOFTWARE
 1501 FF73              ; CALCULATION.
 1502 FF73              ;
 1503 FF73  54 45 52    TABLE     .BYTE 'TER'     "TERMX", WHERE X IS THE TERMINAL ADDRESS
 1504 FF76  00                    .BYTE $00       NULL BYTE ADDRESS
 1505 FF77  53 41 4E              .BYTE 'SAN'     "SANDX", WHERE X IS THE GRAPHIC 7 ADDRESS
 1506 FF7A  00                    .BYTE $00       NULL BYTE ADDRESS
 1507 FF7B  43 41 50              .BYTE 'CAP'     HARRIS BIU
 1508 FF7E  00                    .BYTE $00       NULL BYTE ADDRESS
 1509 FF7F  42 41 43              .BYTE 'BAC'     BACKBOARD
 1510 FF82  BB                    .BYTE $BB       BACKBOARD BYTE ADDRESS
 1511 FF83              ;
 1512 FF83              ; THE STRING TABLE FOLLOWS BELOW.  EACH STRING HAS A SYMBOLIC OFFSET
 1513 FF83              ;
 1514 FF83              WHSYS     = 0             WHICH SYSTEM MESSAGE
 1515 FF83  57 48       ASCII     .BYTE 'WHICH SYSTEM?'
 1515 FF85  49 43
 1515 FF87  48 20
 1515 FF89  53 59
 1515 FF8B  53 54
 1515 FF8D  45 4D 3F    ASCII     .BYTE 'WHICH SYSTEM?'
 1516 FF90              CR_F      = *-ASCII       CARRIAGE RETURN, LINE FEED
 1517 FF90  0D                    .BYTE  $0D
 1518 FF91              LF        = *-ASCII       LINE FEED ALONE
 1519 FF91  0A                    .BYTE  $0A,00
 1519 FF92  00
 1520 FF93              ;
 1521 FF93              LINKT     = *-ASCII       LINK TERMINATED MESSAGE
 1522 FF93  4C 49       .BYTE 'LINK ENDED.'
 1522 FF95  4E 4B
 1522 FF97  20 45
 1522 FF99  4E 44
 1522 FF9B  45 44 2E              .BYTE 'LINK ENDED.'
 1523 FF9E  0D                    .BYTE  $0D,$0A,00
 1523 FF9F  0A
 1523 FFA0  00
 1524 FFA1              ;
 1525 FFA1              LINKR1    = *-ASCII       LINK REQUEST MESSAGE
 1526 FFA1  07                    .BYTE $07
 1527 FFA2  4C 49                 .BYTE 'LINK REQUEST FROM '
 1527 FFA4  4E 4B
 1527 FFA6  20 52
 1527 FFA8  45 51
 1527 FFAA  55 45
 1527 FFAC  53 54
 1527 FFAE  20 46
 1527 FFB0  52 4F
 1527 FFB2  4D 20
 1528 FFB4  07                    .BYTE $07,00
 1528 FFB5  00
 1529 FFB6              LINKR2    = *-ASCII
```

```
CARD # LOC   CODE       CARD  10        20        30        40        50        60        70
 1530  FFB6  2E 20             .BYTE  '. (Y OR N)?'
 1530  FFB8  20 28
 1530  FFBA  59 20
 1530  FFBC  4F 52
 1530  FFBE  20 4E
 1530  FFC0  29 3F
 1531  FFC2  0D                .BYTE  $0D,$0A,00
 1531  FFC3  0A
 1531  FFC4  00
 1532  FFC5              ;
 1533  FFC5              LINKA   = *-ASCII      LINK ACCEPTED MESSAGE
 1534  FFC5  4C 49             .BYTE  'LINK OKAY.'
 1534  FFC7  4E 4B
 1534  FFC9  20 4F
 1534  FFCB  4B 41
 1534  FFCD  59 2E
 1535  FFCF  0D                .BYTE  $0D,$0A,00
 1535  FFD0  0A
 1535  FFD1  00
 1536  FFD2              NOVAIL  = *-ASCII      SYSTEM NOT AVAILABLE MESSAGE
 1537  FFD2  53 59             .BYTE  'SYSTEM NOT AVAILABLE.'
 1537  FFD4  53 54
 1537  FFD6  45 4D
 1537  FFD8  20 4E
 1537  FFDA  4F 54
 1537  FFDC  20 41
 1537  FFDE  56 41
 1537  FFE0  49 4C
 1537  FFE2  41 42
 1537  FFE4  4C 45 2E          .BYTE  'SYSTEM NOT AVAILABLE.'
 1538  FFE7  0D                .BYTE  $0D,$0A,00
 1538  FFE8  0A
 1538  FFE9  00
 1539  FFEA              ;
 1540  FFEA              ; THE FOLLOWING 12-BYTE BIU FUNCTION DESCRIPTION IS PLACED INTO EACH
 1541  FFEA              ; BIU STATUS MESSAGE.
 1542  FFEA              ;
 1543  FFEA  47 52       BIUFNC  .BYTE  'GRAPHIC7 BIU'
 1543  FFEC  41 50
 1543  FFEE  48 49
 1543  FFF0  43 37
 1543  FFF2  20 42
 1543  FFF4  49 55
 1544  FFF6              ;
 1545  FFF6                      *=$FFFA
 1546  FFFA  D8 FE       VECTOR  .WORD NMI      NON-MASKABLE INTERRUPT VECTOR
 1547  FFFC  00 F8               .WORD RESET    RESET VECTOR
 1548  FFFE  B5 FD               .WORD IRQ      IRQ VECTOR
 1549  0000              ;
 1550  0000                      .END


END OF MOS/TECHNOLOGY 650X ASSEMBLY VERSION 5.1
NUMBER OF ERRORS =   0,  NUMBER OF WARNINGS =   0
```

SYMBOL TABLE

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
ABUF	FA12	548	536
ACKOK	F9E4	503	497
ACKCK	F9D5	495	474
ACKBYT	0046	132	1272 1277 1320
AKLOOP	F9B6	473	476
ALLOC	FF39	1446	1186 1388
ALRET	FF42	1451	1448
ANDONE	F9F5	514	515
ASCII	FFA3	1515	1488 1516 1518 1521 1525 1529 1533 1536
ASKAGN	FAF5	733	726 795 1041
BADACK	F9BE	480	499 501
BIUFNC	FFEA	1543	1157
BIUPTR	000A	92	548 550 1098 1101 1106 1123 1137 1141 1144 1147 1150 1158 1392 1394 1399 1404 1407 1410 1413
BUFMEM	0800	78	79
BUFLEN	0080	77	79 259
BUFSTK	0059	161	241 1450 1478
BUFCNT	0016	79	147 161 165 166 237 247
BUFFRI	F841	254	263
BYTLOW	0041	127	595 631 662 691
BYTES	FA90	659	652 788
BYTRET	FA96	663	676
CHARS	FA7F	647	640 653
CHKLOP	FE5B	1299	1301
CKCRET	F902	1070	1095
CKETTY	FCEE	1058	1046 1049
CKSTAT	FCF6	1067	1059 1062
CKTOUT	FC9E	1009	355
CKYOUT	FC07	1045	1010 1013 1031
CLOOP	FEFF	1399	1401
CNCT	0002	71	272
COLIDE	FDE6	1234	1227
CONMAT	F80F	753	751
CONECT	004E	141	287 327 328 529 638 792 806 837 911 934 942 960 977 1045 1050
CONFRE	F8FF	917	913
CONOUT	F8CA	874	867
CONDS	F8C0	868	865
CONSOV	FC8C	1029	1036
CONTRM	F829	767	761
CONYES	F876	811	809
CRLF	000D	1515	704 793 811
CTIME	FD03	1078	1012 1048 1061 1068
CTRET	FD17	1088	1080 1081 1084 1085
CURNET	000C	93	1192 1347
CURBIJ	000E	95	294 552 778 804 1165 1395
CURDEV	000D	94	553 680
DATMSG	0000	12	534 835
DMATM	0042	128	578 611
DMAMOD	FA59	610	604
DMATTY	0068	156	232 543 571 580 602 609 670
DOCNCT	F89A	270	274
DQ	FF51	1471	510 983
DUARTS	1400	23	1143

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
ENQ	FF43	1456	295 681 779 805 1166 1348
ENQ1	FF45	1458	1460
FFLOOP	F864	233	235
FORJS	FE14	1260	1254
GCRLF	FACD	704	698
GETDAT	FA4F	602	564
GETRPY	FA84	691	639
GETCH	FA24	561	533 544 579 654
GETYN	F847	786	641 789
GETBYT	FE34	1279	1293 1297
GETCH1	FA0C	542	530 531 700
GETIT	FE2C	1275	1263
GLF	FAC9	702	696
GPUT	FACF	705	703
HARGO	004D	140	284 301 307 922
HARSOV	F8D0	314	752
HARBAS	0021	6	288 305 317 918 920 1019 1021 1035
HARSOR	001F	8	270
HIPTR	0095	165	248 256 404 831 1190 1393
HOLDTX	F9A0	457	458
INBC	0005	88	556 561 647 651 659 693 699 734 931
INDONE	FA97	669	587 594 597 599
INDON1	FA9F	674	650
INDEV	F9FA	529	353 683
INLOOP	FE57	1296	1294
INPTR	0003	87	549 551 648 678
INRET	F846	781	802
INSET	000F	96	532 554 682
INTSET	0011	98	1184 1193 1262 1349
INTFLG	0040	126	394 418 1357
INTRES	FD83	1194	1185 1187
INTBC	0002	86	1214 1239 1242 1260 1295 1296
INTX	003F	126	1215 1355
INTBUF	FD9C	1182	342 352 1353
INTPTR	0000	85	1189 1191 1276 1288 1331 1335 1340 1343
IRET	FA0B	537	572 600 634
IRQ	FD85	1203	1548
IWR	0002	57	562
KEYNOF	F9A8	462	463
LASTTX	003B	121	1332 1341
LASTS2	003C	122	1336 1344
LF	000E	1519	702
LINKR2	0033	1529	974
LINKT	0010	1521	334 925
LINKA	0042	1533	813 950
LINKR1	001E	1525	965
LOPTR	007F	165	250 254 257 402 829 1188 1391
LWAIT	F91A	378	387
MATCH	F803	748	715
MAXPAX	0018	104	210 649
MLOOP0	F900	343	323
MLOOP	F901	350	356
MSGOK	FEAD	1339	1333
NETBSY	F962	422	397
NETRET	F9F9	516	487
NETFRE	F9EA	509	490
NET	F916	364	350

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
NEXT	0050	147	293 509 679 777 803 982 1118 1131 1164 1346 1459 1461 1462 1473 1474 1475
NEXTCH	FAD6	711	718
NEXTEN	FAD4	710	728
NICLRX	FECA	1354	1232 1251 1258 1337
NIRET	FECC	1355	1222
NMATCH	FAE5	719	713
NMBCRC	0032	115	1311
NMCOL	002C	112	436 1249
NMDISC	002E	113	488
NMGCRC	0030	114	1350
NMI	FED8	1372	1546
NMLOST	0034	116	1264
NMRXMT	002A	111	482
NMWAIT	0036	117	422
NMXMT	0028	110	504 1105 1173
NOSOF	F8F4	338	302 333
NOSYS	FCCD	1038	1020 1026 1034
NOTPWR	F8E5	327	286
NOVAIL	004F	1536	1038 1051
NOXMIT	003E	124	378 381 500
NOZERO	F850	790	787
NRET	F967	424	365
NTPRX	0039	119	409 484 485 512
NTXCOLI	F97A	439	435
NUARTS	0C00	21	188 190 395 419 430 439 451 460 473 1140 1212 1231 1235 1257 1267 1280 1285 1299 1304 1308 1515 1319 1326 1374 1376
NUARTO	0C01	22	443 455 495 1213 1287 1306 1321
NULRET	FBCE	876	904
NULL	FBEA	903	872
OUTFRE	FC7D	982	840 854 917 935 954 956 959 962 964
OUTDEV	FBB1	825	351
OUTLOP	FBDF	894	898
OUTBC	0008	90	414 446 470 496 846 850 878
OUTPTR	0006	89	403 405 408 411 413 442 469 830 832 834 847 868 874 916 947 968 1226
OUTSET	0010	97	825 844 853
OUTPL	0009	91	848 851 871
OWR	0002	64	896
P1ACR	101B	30	192
P1ADDR	1013	27	194
P1BDDR	1012	26	195
P1IFR	101D	32	563 991 1146
P1IER	101E	33	206
P1PCR	101C	31	200
P2ACR	102B	38	197
P2ADDR	1023	37	212
P2BDDR	1022	36	213
P2IFR	102D	40	897
P2IER	102E	41	207
P2PCR	102C	39	202
P3ADDR	1043	45	214
P3ACR	104B	46	198
P3BDDR	1042	44	196
P3IFR	104D	48	273
P3IER	104E	49	208

SYMBOL	VALUE	LINE DEFINED	CROSS-REFERENCES
P3PCR	104C	47	204
PARITY	0038	119	392 444 445 454 1278 1290 1291 1309
PARERR	FE77	1311	1286 1305
PCONST	FEE9	1387	292 535 773 801 1094
PDATAJ	FC3A	949	940
PDATA	F8A2	844	839 949
PLINKA	FC3D	950	948
PORT1B	1010	24	621
PORT2B	1020	34	885
PORT3B	1040	42	265 314 764 1253 1406
PORT3A	1041	43	216 271
PORT2A	1021	35	886
PORT1A	1011	25	629
PRET	FF19	1414	1390
PREPAR	F92F	392	384
PRIMER	0019	9	215
PUTCH	F8AC	850	826 879
PUTSTR	FF63	1487	335 341 705 738 794 812 814 926 928 951 966 975 1039 1052 1054 1493
PUTRET	FF72	1494	1489
QIN	0066	151	232 233 827 982 1126 1346
QIT	F83D	777	774
QOUT	0067	152	293 364 509 679 777 803 1113 1164
RAMSIZ	0C00	76	78
RANDU	0012	99	266 1236 1244 1247
RESET	F800	184	775 923 1547
RNDCNT	0013	100	382 383 385 1248
RSXADR	FCC5	1033	1022
RXDATA	FE07	1253	1217
SACMSG	00DF	15	800 937
SBOT3	FD5D	1136	1127
SBOT2	FD49	1123	1114
SFINC	FF2C	1435	423 457 483 489 505 1250 1265 1312 1351
SFRET	FF38	1441	1436 1438
SKIPI	F88E	262	260
SKIPIT	FE0C	1256	1261 1284 1303 1313
SLOOP1	FD2D	1105	1110
SLOOP2	FD3D	1115	1122
SLOOP5	FD96	1173	1175
SLOOP3	FD51	1128	1135
SLOOP4	FD79	1157	1162
SNDACK	FE7E	1315	1273 1310 1317
SNDOUT	F8D7	885	876 973 1491
SNDSOF	F884	291	306 336
SNDSTA	FD18	1093	1069
SNDLT	FC0D	925	919 921
SOFMSG	00DE	14	291 909
SONMSG	00E0	16	772 958
SONCNT	0044	130	944 1023 1024 1033 1040
SQNUM	003A	120	406 513
STACKI	F86E	240	243
STAMSG	00DB	13	1093
STIMER	FF1A	1420	620 771 980 1169
STKPTR	003D	123	238 1446 1449 1476 1479
STRING	0047	133	694 711 759
SWAIT2	F928	385	386
SWAIT1	F924	383	379

```
SYMBOL  VALUE  LINE DEFINED        CROSS-REFERENCES

SYSTMX   F92C        769   322  756 1030
TABLE    FF73       1503   712  754
TERMX    F816        758   749
TERMF    0045        131   810  864  932  955
TEST2    FDD4       1225  1229
TESTTX   FDD2       1224  1221
TETTY    0025        109   618 1060
TICK     004F        143   221  994  997
TIMRH    1015         29   219
TIMRL    1014         28   217  993
TIMSET   FC9D       1003   992  995  999 1001
TIMOUT   FC83        991   354
TOD      0019        105   998 1000 1002 1078 1082 1086 1421 1423 1426
TOKEN    0015        102   720  722 1218 1225
TRMCNT   0008          7   305  920 1021 1025
TRNSMT   F9B8        429   421
TRYDS    FB87        863   852
TRYSOF   FBEE        909   836
TRYLAT   FBE9        899   877  895
TRYSAC   FC24        937   910
TRYSON   FC4C        958   938
TSACK    001F        107   770 1011
TSTAT    001C        106   268 1067 1168
TTYOP    FA3F        593   573
TTYMOD   FA5D        618   603
TTY      0002         17   534  772  835
TTYTM    0043        129   588  593  672 1058 1063
TXBEMP   F970        434   441
TXCOL    F974        435   450
TXLOOP   F96F        433   447
TXNACK   FEC7       1353  1328
TXPAR    F98E        449   453
TYACK    0022        108   979 1047
VECTOR   FFFA       1546  ****
WAIT75   FE91       1323  1324
WAIT     0017        103   633  735  780  939  941  963 1009 1029
WAITBY   FE35       1280  1282
WAT100   F956        416   417
WHSYS    0000       1514   340  737  927 1053
XADDR    004C        139   223  289  303  304  316  319  339  736  763  766
                              769  807  914  930  952  969 1018 1149 1403
XMIT     0014        101   431  434  449  462  471  480  514 1216 1378
YES      F85F        800   791
YNRET    F880        815   828
ZLOOP    F85D        226   228
```

REFERENCES

[1] Benbow, R. L., et al., Advanced Crew Activity Planning System Crew Activity Planning System Conceptual Design Report, McDonnell Douglas Corporation, MDC W0020, 4 March 1977.

[2] Allen, M. A., Final Briefing CAPS Communication Interface Study, The MITRE Corporation, 11 December 1978.

[3] Graphic 7 Computer Graphics Display System Graphic Support Software (GSS-4) Version 1.1 User's Manual, Sanders Associates, Inc., Nashua, New Hampshire, March 1978.

[4] Graphic 7 Computer Graphics Display System Technical Description, Sanders Associates, Inc., Nashua, New Hampshire.

[5] Graphic 7 Computer Graphics Display System Programmer's Reference Manual, Sanders Associates, Inc., Nashua, New Hampshire, February 1978.

[6] Standard Graphics Control Program (GCP) Source Listings Release 2, Sanders Associates, Inc., Nashua, New Hampshire.

[7] Pruett, D. M., Harris - Sanders Parallel Interface Handler *V:SANH:V Software Definition Document, NASA Institutional Data Systems Division, JSC # ,December 1979.

[8] Herndon, E. S., Letter to J. L. Fisher, Subje : Suggested Modifications to the Crew Activity Planning System Host Control Program, The MITRE Corporation, D72-L-383/HO, 23 February 1979.

[9] Hopkins, G. T., A Bus Communications System, The MITRE Corporation, MTR-3515, November 1977.

[10] Brown, J. S. and G. T. Hopkins, TMS Communications Hardware Volume II - Bus Interface Unit, The MITRE Corporation, MTR-4721 (JSC #14723), March 1979.

REFERENCES

(Continued)

[11] Allen, M. A. and G. S. Roman, Specification for Installation of the Crew Activity Planning System Coaxial Cable Communication System, The MITRE Corporation, MTR-4726 (JSC #14869), April 1979.

[12] Allen, M. A. and G. S. Roman, Crew Activity Planning System Coaxial Communications Cable, The MITRE Corporation, MTR-4732 (JSC #16316), December 1979.

[13] Brown, J. S. and M. D. Lenker, Diagnostic Procedures for Trend Monitoring System (TMS) Communications, The MITRE Corporation, MTR-4724 (JSC #14794), March 1979.

[14] Technical Manual Universal Block Controller (UBC) Input/Output Channel, Harris Corporation, Fort Lauderdale, Florida, September 1978.

[15] (Preliminary) Graphic 7 Parallel Interface Circuit Card Assembly, Sanders Associates, Nashua, New Hampshire, March 1979.

[16] Graphic 7 Computer Graphic Display System Terminal Controller Technical Manual Volume 2 Maintenance Diagrams, Sanders Associates, Nashua, New Hampshire, May, 1978.

[17] Herndon, E. S., Letter to J. L. Fisher, Subject: Documentation of Modifications to the Sanders Graphic 7 Parallel Interface Card to allow Teletype Operation with the Harris Slash 8 Computer, The MITRE Corporation, D74-L-39/HO, 19 November 1979.

[18] Wagner, P. E., RF Modem for CATV Listen-While-Talk Bus Interface Unit, The MITRE Corporation, MTR-3572, April 1978.

[19] "MCS 6522 Versatile Interface Adapter Preliminary Data Sheet," MOS Technology, Inc., Norristown, Pennsylvania, March 1977.

[20] "MC 6850 Asynchronous Communications Interface Adapter (ACIA) Data Sheet," MOTOROLA Semiconductor Products, Inc., Phoenix, Arizona.

REFERENCES

(Concluded)

[21] VULCAN I/O Services, Harris Corporation, Inc., Fort Lauderdale, Florida, December, 1977.

[22] Gregor, P. J., TMS Communications Software Volume II - Bus Interface Unit, The MITRE Corporation, MTR-4723 (JSC #14793), April 1979.

[23] MCS 6500 Microcomputer Family Programming Manual, MOS Technology, Inc., Norristown, Pennsylvania, January 1976.