AUTOMATIC COMPUTER DATA BASE ENTRY AND SETUP OF A DECOMMUTATION SYSTEM

M. Kimberley Rogers
James H. Ashmore
General Dynamics Convair Division
San Diego, California

Gina L. Lamb
Richard A. Schuh
Loral Instrumentation
San Diego, California

I. Abstract

This paper presents a method developed to automate the data base entry and setup of the ADS 100 Decommutation System. Automation was accomplished by interfacing an existing RS-232C port with a VAX computer. Other available interface options are considered. The automated system provides a method for rapid data entry while minimizing errors. Automation also eliminates the continuing requirement for a skilled ADS 100 programmer. Additional topics reviewed are the various problems encountered while developing the interface. Also discussed is the development and interface of host computer software, including the predefined ADS 100 record structures. The final result is a complete and accurate digital data base setup in the ADS 100 system.

II. Background

In the past several years, telemetry commutation and decommutation has changed dramatically. Equipment that is five to ten years old can be archaic and monotonous to use. Retrieval of data may be slow and its correlation can be expensive. In contrast, today's equipment is fast and can process considerably more data that its predecessors could. However, with this increased processing power has come increased programming requirements to enable the real-time decommutation of analog and digital data. Literally thousands of parameters may need to be defined to set up one system, requiring many manhours and a good knowledge of the particular system.

Many telemetry systems still record the decommutated data and leave the processing for mini or mainframe computers. Typically, the data may be stored on nine-track tapes and processed at some future time. Applications today frequently require immediate processing and the capability of controlling and updating the actual decommutation of a data stream. An ideal situation is to have the decommutation system interfacing directly with a host computer. This allows a means for quick, automatic configuration and setup of the data and associated decom files. A modern decommutation system will then

convert and format received data into engineering units for use in controlling and monitoring a test operation in real time.

III. Introduction

Automating the data base entry and setup of the decommutation system is desirable for several reasons. Manually formatting and entering a large amount of data is time-consuming and likely to introduce errors into the system. Utilizing a host computer to directly and automatically download necessary information is a highly desirable feature. The initial data base is created and stored on the host computer. Software developed on the host performs reformatting peculiar to the decommutation system. By transmitting this preformatted data interactively, completeness, accuracy, and time efficiency are maximized.

The Advanced Decommutation System Model 100 (ADS 100), built by Loral Instrumentation, is a highly flexible real-time decommutator and processor. This system accepts output from a receiver and provides the capabilities to distribute, limit-check, compress, convert, and display the data. The ADS 100 provides two direct host computer interfaces, the RS-232C and the General Purpose Interface Bus (GPIB).

This paper presents information on the automated downloading process developed at General Dynamics Convair Division. For this application, software developed on a VAX computer manipulates previously established PCM data bases so that they are in a format acceptable to the decom system. The ADS 100/VAX interface became necessary due to the deficiencies discovered in using a manual system. There is a requirement to load multiple large and varying data bases. Manual entry introduced typographical errors in data records and setup information, causing incorrect or incomplete configurations within the system. A skilled ADS 100 programmer was necessary to enter and verify all records. This became a costly and inefficient method of loading the data.

General Dynamics presently uses the ADS 100 in a variety of ways. It serves as a PCM monitor of test vehicle health, both during vehicle checkout and real-time testing. FM data is digitized and treated as PCM with the ADS analog-to-digital conversion function. It is also used in post-test analysis of analog and digital data. The following discussion concerns the development of a process to download PCM data bases automatically into the ADS decommutation system, where the system will function as a monitor and analyzer of actual data.

IV. Interface Selection

Several methods of interfacing between the decom system and the host computer are available, and each has its relative advantages and disadvantages. The first method considered was to develop all data bases and respective system setup information on the VAX and then load them to and store them on nine-track digital tapes. The information from the tapes could then be read into the ADS when needed. This approach requires no additional peripheral hardware for the VAX (nine-track tape drives being readily available) and would utilize existing locally-developed subroutine libraries. However, some additional hardware would be required for the communication link. Specifically, the ADS would need a digital tape interface board and associated nine-track tape drive. Current system software would transmit the information from the tape to the memory of the machine. From here, it would ultimately be transferred to and stored on three-inch diskettes.

Another method available was to attach a diskette drive to the VAX and create ADS 100 compatible diskettes. This method presents the difficulty of finding an appropriate drive. Additional complications were the complexity of developing software on the VAX to support the drive and a problem with duplicating the diskette format. If discs were created directly on the VAX, software maintenance of the disc would be required. This is due to inevitable system software updates, possibly requiring a change in the record structure on the diskette.

A third method considered was the General Purpose Interface Bus (GPIB). This allows a direct, high-speed connection between the ADS and the VAX. A disadvantage to this method is that the system must be within 100 feet of the host computer. For the application developed at Convair Division, the ADS and host computer reside in separate buildings. Hence, this interface would have required transporting the system to the VAX whenever a transfer of information was needed.

The final method available is the serial RS-232C communication line. Utilizing this interface, the ADS 100 functions as a VAX terminal with a bidirectional data link to the VAX computer. Unlike the GPIB bus interface, there is no limit on the distance between the decom system and host computer. Modems are used for both machines, and communications occur over the Local Area Network (LAN) telephone lines. This method requires a minimum of additional hardware, only two modems and associated cables. It also requires the development of little additional software beyond the baseline programming necessary for all methods. This interface was selected for the automated process discussed. The RS-232C communication line allows bidirectional, interactive transmission of the data, including complete operational setup information and decommutation data. These four methods are depicted in Figure 1, respectively.

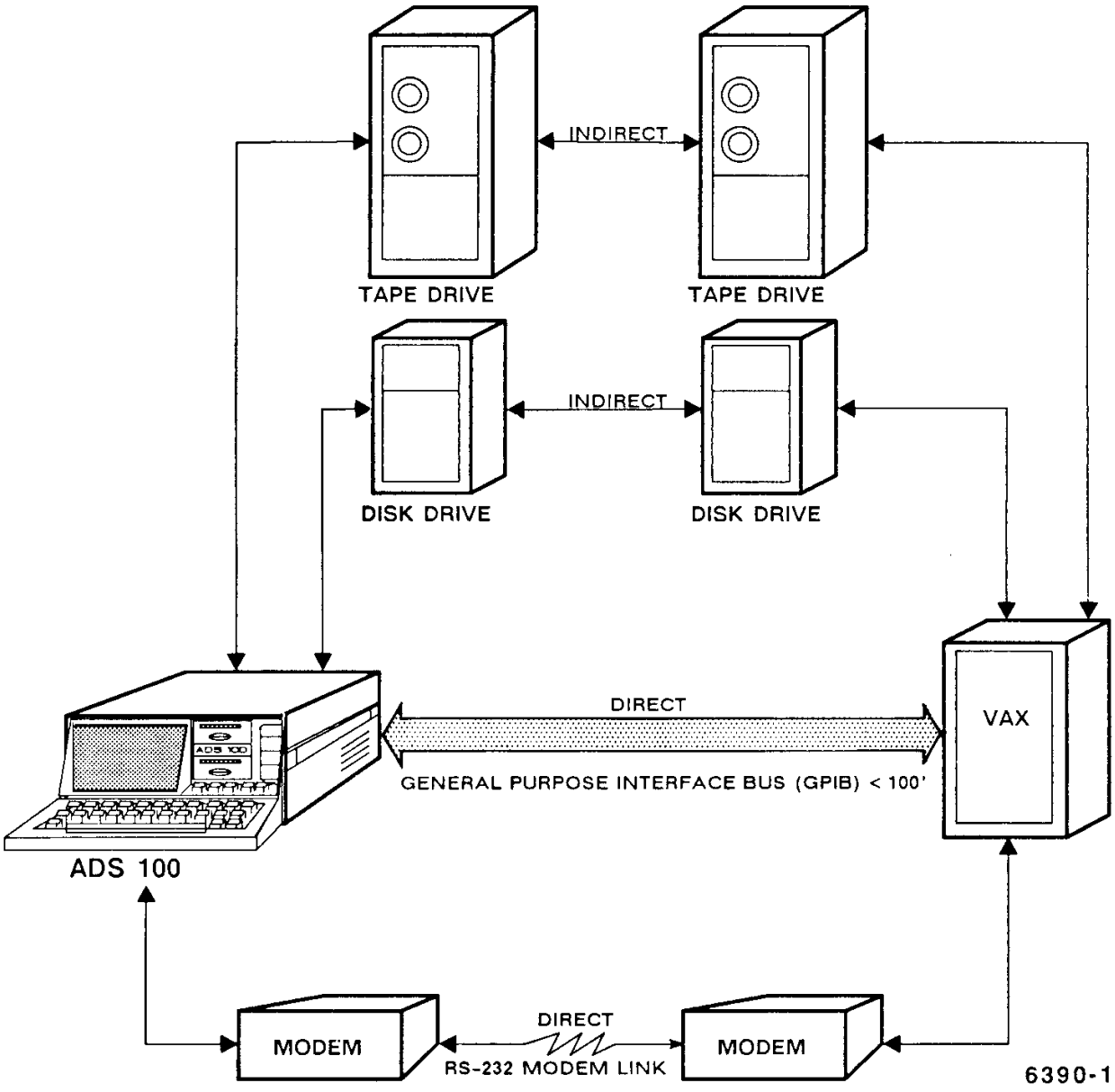


Figure 1. Available interface options.

V. Record Structure and Protocol

The ADS 100/VAX RS-232C interface entails passing ASCII and binary strings (or records) between the two machines. The strings passed are encoded using data link communication protocol. This standard protocol scheme helps to ensure the integrity of the data, which is often compromised in RS-232C asynchronous communications. Each string is transmitted as a series of individual asynchronous characters. Four protocol characters and a block check character (BCC) must be transmitted along with each record. These ASCII characters are shown in Figure 2 in the sequence required to encode each record.

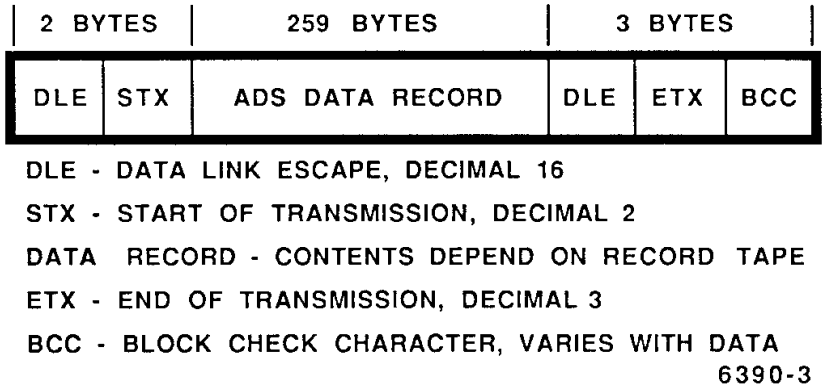


Figure 2. ADS RS-232C record protocol.

The BCC is a single byte variable whose value is determined by Exclusive-Oring (.XOR.) of all the bytes in the data record and the ETX of the terminating sequence. The receiver (either the host or the decom system) verifies the BCC to check the integrity of the transferred data. It then evaluates the sequence of the protocol characters to determine boundaries of records. A further protocol rule is applied when the DLE character value exists as data within the record. Another DLE character or 16 must be inserted adjacent to the data value 16. This informs the system that the byte is not a control character, but a part of the data string.

Two primary types of records are passed between the VAX and the ADS: file records and data records. A file record is sent as a header or marker for the system or module file to follow. File records have fixed values, which determine the hardware/software module used within the ADS. The ADS responds to file records with data records, which contain the requested information. Data records have variable binary and ASCII values representing the data base setup information for the respective file.

A typical data transfer consists of the VAX sending over a single file record and multiple data records to set up the ADS Parameter module. First, the file record is sent to the system, indicating that the data records to follow will contain Parameter information. The file record has informed the system of the type of transfer taking place; then the operating system of the ADS will load the data records sequentially to appropriate memory locations. Once data base and setup information has been downloaded, it may be read from or updated via the ADS. In addition, real-time data may be read or sent back to the host computer for further processing. Similarly, when transferring information back to the VAX, a file record with the appropriate file record structure will be sent to the decom system.

This serves as a request for data from a particular module. The system will return the file record, marking the transfer, followed by the data records containing the requested information. Uploaded data may then be read, modified, retransmitted, or stored by the host.

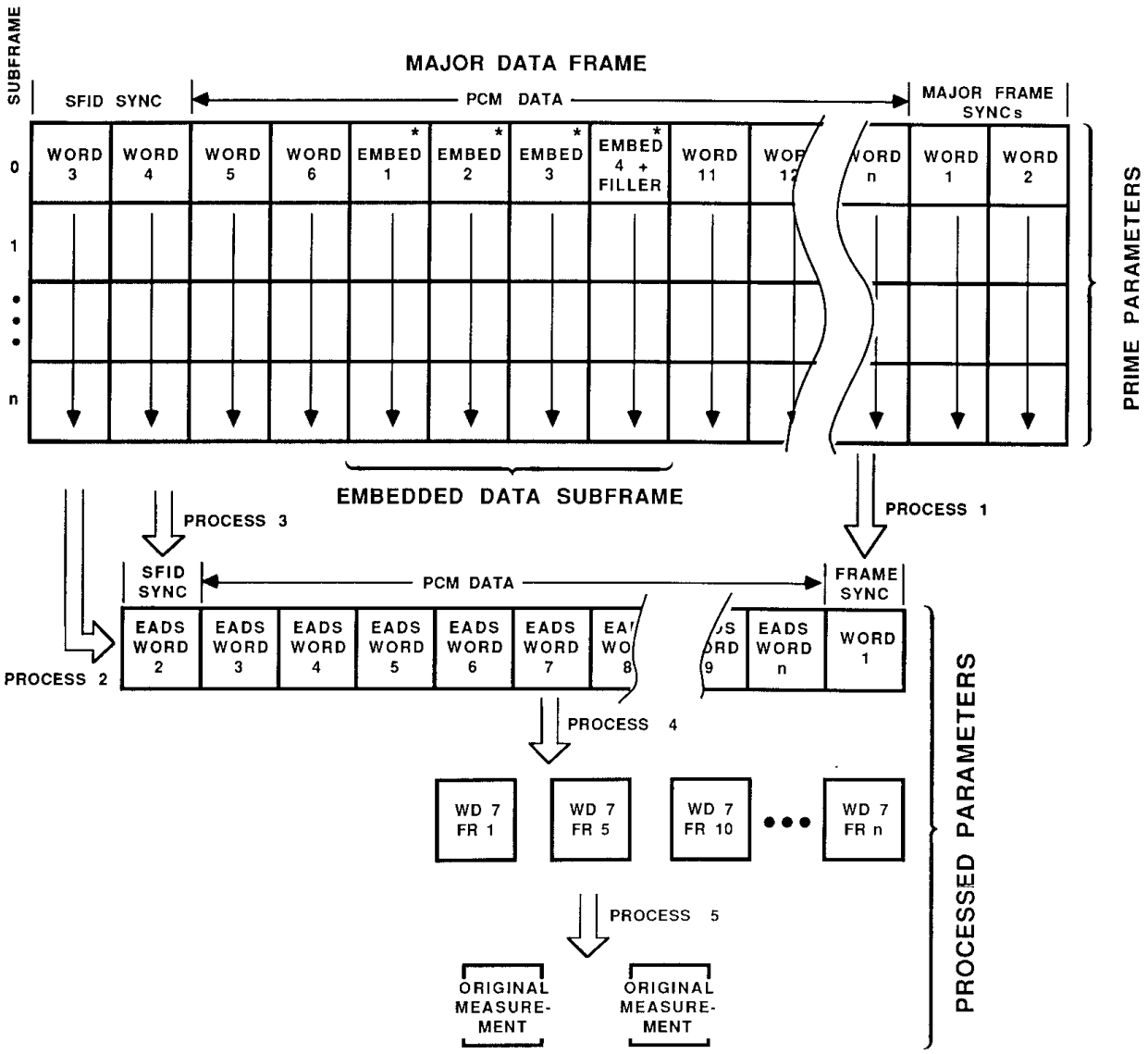
Throughout these transfers, the receiver continually verifies the record protocol of each record. In the primary application of downloading from the VAX, an acknowledgement (ACK, decimal 6) reply is returned to the host computer CRT when a valid data record is received (i.e., the BCC value and the record structure are correct, and the data field types are acceptable). If this process locates an invalid record, a negative acknowledgement (NAK, decimal 21) will be sent to the VAX and appropriate action will be taken (either retransmission attempts or the execution of an error message). This same handshaking scheme occurs internally when the VAX receives data from the ADS.

VI. VAX Program Development

For this application, the host software was developed using the VAX FORTRAN language (an extension of FORTRAN 77). This language was selected due to its availability and its familiarity to potential users. The driver program incorporates the ADS 100 text format and specific I/O RS-232C interface formats. For communications, a VAX system service routine called QIO (queued I/O) was utilized. This service simplified communications while allowing asynchronous communications and timed responses. QIO queues an I/O request to another device (e.g., a terminal, tape, or disc drive). For this application, the ADS is effectively a VAX terminal device.

The decommutation system is configured in a data-driven distributive processing architecture. Various hardware modules that perform data recognition, timing, and distribution must be set up. The data base and setup information are primarily configured by defining parameters. A parameter either identifies where in the serial stream the data is found or defines a process to be performed on a source parameter. Processing algorithms for parameters include frame and subframe identification (sfid) distribution and synchronization, bit compression, bit swap, etc. Parameters are passed as sources from one to another until all processing on the original data word is complete. Figure 3 depicts this system structure. In real time, the decom system accepts a data stream and separates it into its original measurements. The sequence of execution within the machine depends on the order of the data received.

The driver program was designed around the system structure of the decom. Each functional file (Parameter, Decom, etc.) is broken into a separate subroutine. Each of these subroutines creates parameters consisting of the necessary file record and all associated data records. This modular style programming allows subroutines to be



*THE EMBEDDED ASYNCRONOUS DATA STREAM (EADS) IS SUBCOMMUTATED INTO WORDS 7 THROUGH 10- ALL SUBFRAMES OF THE MAIN DATA STREAM. EMBED WORDS ARE PRIME SOURCE PARAMETERS FOR EADS WORDS. EADS WORDS ARE PROCESSED SOURCE PARAMETERS FOR WD_FR_ WORDS, ETC.

PROCESS 1, FRAME SYNCRONIZATION: PERFORMS BIT RECOGNITION TO IDENTIFY EMBEDDED FRAME SYNCRONIZATION WORD(S). THIS, IN TURN, IDENTIFIES WORD BOUNDARIES WITHIN EMBEDDED MINOR FRAMES.

PROCESS 2, FRAME DISTRIBUTION: ALLOWS SELECTION OF A SPECIFIC WORD LOCATION WITHIN THE EMBEDDED MINOR FRAMES.

PROCESS 3, SUBFRAME SYNCRONIZATION: ALLOWS IDENTIFICATION OF THE CURRENT EMBEDDED MINOR FRAME NUMBER. THIS (TOGETHER WITH SUBFRAME DISTRIBUTION) CAPTURES SUBCOMMUTATED DATA WITHIN THE EADS.

PROCESS 4, SUBFRAME DISTRIBUTION: ALLOWS SELECTION OF A SPECIFIC SUBCOMMUTATED DATA WORD WITHIN THE EADS, BY WORD AND SUBFRAME LOCATION.

PROCESS 5, BIT COMPRESSION: CAPTURES ALL SAMPLES OF A SPECIFIED SOURCE PARAMETER AND COMPRESSES THE BITS BY LEFT-JUSTIFYING BITS OF INTEREST. PROCESS 5 COULD USE ANY OF VARIOUS ALGORITHMS THAT MANIPULATE BITS IN SOURCE DATA WORDS TO FORM NEW DATA WORDS. 6390-4

Figure 3. Sample decommutation system structure.

developed and tested independently and gives user control in loading specific types of data. Difficulties in the software development occurred as a result of the specific, complex requirements. It was necessary to allow for three separate and different formats of prime (primary bitstream) data, depending upon the main bitstream rate. This resulted in a requirement for three versions of the Decom module, which set up bit synchronization information. Additionally, there are three different embedded PCM data formats, depending upon the type of test vehicle. The prime and embedded data streams may be combined interchangeably in the various data bases. The Parameter system file defines the complete PCM data stream.

To simplify the program development, the Parameter system file was divided into two major sections, the PCM prime data and the embedded data. Each section was then divided into three subsections or subroutines. This allowed the special requirements of each format to be handled separately. Each of these three subsections was then further divided into several sections that handle different types of PCM words properly and separately. One section creates words or parameters that include all data bits in their respective PCM addresses. Another section creates source parameters for words that require the bit compression process (i.e., those parameters consisting of less than the full data word). Still another routine creates the original measurements, which use only select bits (e.g., a one-bit discrete). Frame and subframe syncronization parameters, as well as other special-case source parameters, are also created.

In addition to the routines that format the data stream, the program contains routines that configure the decom system. The Function Keys file defines one or more of the available 14 ADS programmable function keys. The Sort function defines parameters to be collected for displaying or printing, according to a specific user-defined event. The I/O Page file sets up communication ports, transmission characters, and printer type. Real-time displays may also be automatically defined; however, for this application they are manually set up in accordance with the individual test. Figure 4 shows the flow of the primary system and module files in the VAX driver program.

VII Problems Encountered

There were many growing pains throughout development of the RS-232C communication link. The following paragraphs highlight some of the more significant problems and how they were overcome.

Before the link could be developed, the system setup had to be checked to ensure the hardware was properly configured and the system RS-232C communications setup was compatible with the modems used. It was decided to configure the communications line to the default settings used on the ADS so that setup actions would not be required

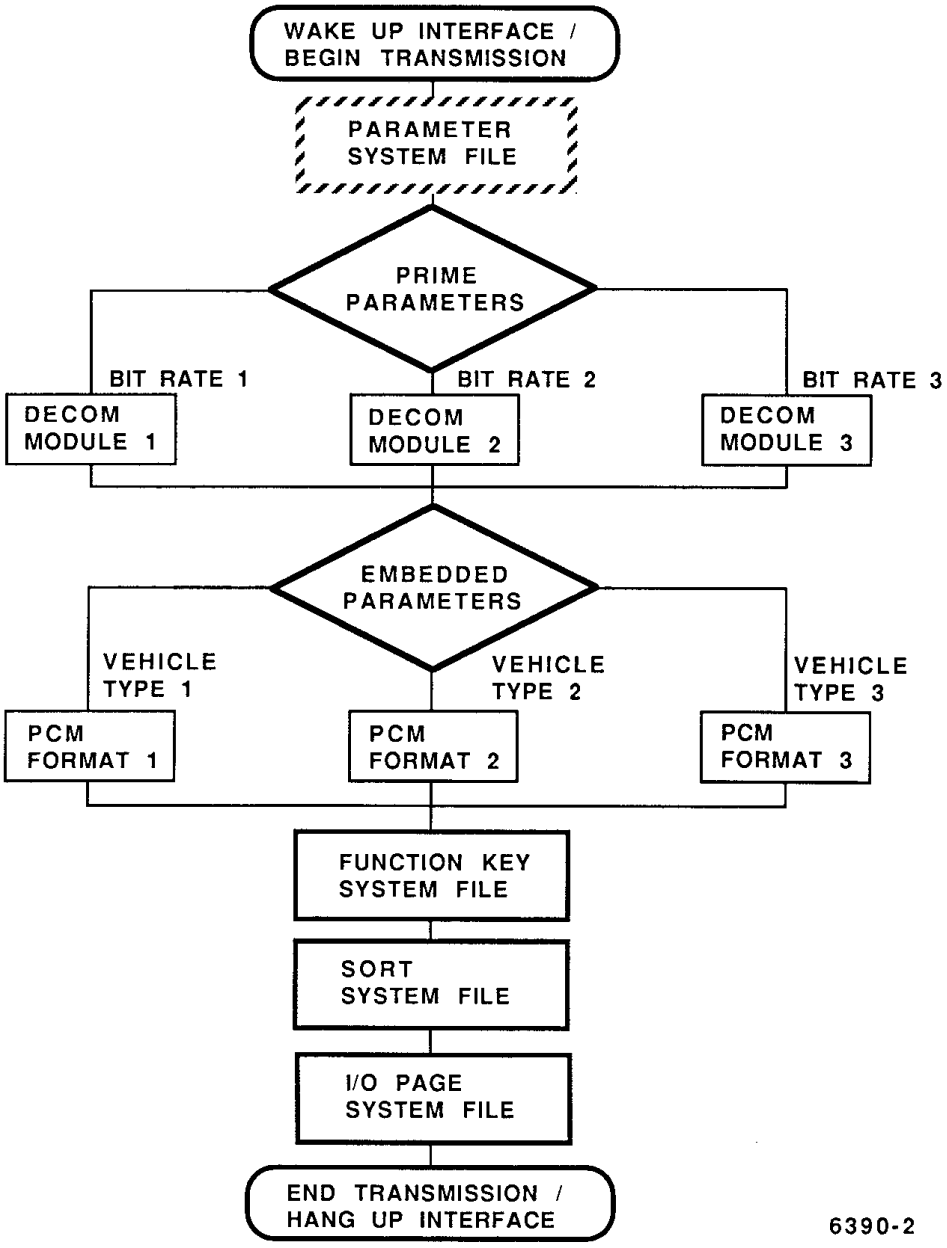


Figure 4. Driver program flow diagram

before each data base transfer. These transmission character settings are 4800 baud, 8 data bits per ASCII character, 1 stop bit, and no parity.

One of the first problems encountered was with the VAX terminal driver erroneously interpreting binary and ASCII characters passing through the link. This interpretation was eliminated by using the terminal setup parameters PASSALL (prevents any interpretation by the terminal driver) and NOWRAP (prevents insertion of carriage controls after every 80 characters).

A similar problem experienced involved the carriage control at the end of every record transmitted. The software naturally sent a carriage return/line feed prior to writing each new record. The ADS tried to interpret these characters. Therefore, it was determined that all carriage control had to be eliminated. This was accomplished by properly configuring the system I/O portion of the VAX software that handles the communications, eliminating carriage control.

Another problem occurred when an attempt was made to pass invalid records. The ADS checks all incoming records to see that acceptable values are in all fields. While this check does not catch all possible errors, it does ensure that the record can be displayed by the system CRT. The NAK response is received for RS-232C transmission-induced errors; hence, it can lead to confusion. A way to ascertain where the trouble lies is to retransmit the record several times. If it is truly a case of bad record structure or content, it will continuously be rejected. If it is a result of intermittent bad communications, the record should be accepted on retransmission.

VIII. Results Obtained

The results obtained more than justified the labors of developing the ADS 100/VAX communication link. The time required for a full data base entry and setup decreased from three days to less than thirty minutes. Errors in the data base became virtually nonexistent. Remaining errors could be traced to the original VAX data base and would have affected manual entry as well. Another goal accomplished by automation was removal of the requirement for a skilled ADS programmer. Once the necessary software was written, the information was easily downloaded interactively. The result was a complete and accurate data base and system setup.

References

1. Stone, Harold S., *Microcomputer Interfacing*, University of Massachusetts, Addison Wesley, 1982.

2. Hewlett-Packard, *Basic Interfacing Techniques*, Hewlett-Packard, Fort Collins, CO, 1984.

3. Loral Instrumentation, *ADS 100 Advanced Decommutation System User's Manual*, Loral Instrumentation, San Diego, CA, 1985.

4. *VAX/VMX System Services Reference Manual*, AA-Z501A-TE, Digital Equipment Corporation, Sept. 1984.

5. *VAX/VMS I/O User's Reference Manual: Part 1*, AA-Z60OA-TE, Digital Equipment Corporation

6. *VAX FORTRAN User's Guide*, AA-D035D-TE, Digital Equipment Corporation, Sept. 1984.

7. *Programming in VAX FORTRAN*, AA-D034D-TE, Sept. 1984.

8. *Guide to Programming on VAX/VMS (FORTRAN edition)*, AA-Y503A-TE, Sept. 1984.