

RPI TECHNICAL REPORT MP-72

A HIGH SPEED TELEMETRY DATA LINK
FOR AN AUTONOMOUS ROVING VEHICLE

by

David John Cipolle

A Study Supported by the
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

under

Grant NSG-7369

and by the

JET PROPULSION LABORATORY

under

Contract 954880

School of Engineering
Rensselaer Polytechnic Institute
Troy, New York

August 1980

CONTENTS

	page
LIST OF FIGURES.	v
ACKNOWLEDGEMENT.	vi
ABSTRACT	vii
1. INTRODUCTION	1
2. SYSTEM CONSTRAINTS	3
2.1 Data Sources.	3
2.2 System Requirements	3
3. TELEMETRY TRANSMITTER.	8
3.1 Data Priority Network	8
3.1.1 Description.	8
3.1.2 Design and Operation	10
3.2 ADLC Operation.	12
3.3 Hardware Description.	16
3.3.1 Initialization PROM Coding	17
3.3.2 Transmitter Sequencer.	18
3.4 Microprocessor Control.	22
3.4.1 Software	26
3.5 RF Transmitter.	27
3.5.1 Hardware Description	29
3.6 Physical Layout	31
4. VEHICLE MULTIPLEXER.	33
4.1 Hardware Description.	33
4.1.1 FIFO Rate Buffer	36
4.1.2 Sequencer.	38
4.2 PROM Format	41
4.3 Physical Layout	44

	page
5. TELEMETRY RECEIVER.	46
5.1 ADLC Configuration	46
5.1.1 Initialization Coding	48
5.2 Hardware Description	48
5.2.1 Controller.	50
5.3 GPIB Signals	54
5.4 R.F. Receiver.	55
5.4.1 Front End	55
5.4.2 Demodulator	56
5.4.3 Antennas.	58
5.4.4 Location.	58
6. PERFORMANCE	60
6.1 Performance Calculations	60
6.2 Diagnostics.	62
7. CONCLUSIONS	64
8. REFERENCES.	67
9. SCHEMATIC DIAGRAMS.	68
10. APPENDICES.	82

LIST OF FIGURES

	page
Figure 1. Timing Constraints	5
Figure 2. Telemetry Frame Format	6
Figure 3. Overall Block Diagram	9
Figure 4. Priority Network Timing Diagram	11
Figure 5. Priority Network Flow Chart	13
Figure 6. Word Format	19
Figure 7. Telemetry Transmitter Block Diagram	20
Figure 8. Telemetry Transmitter Timing Diagram	23
Figure 9. Telemetry Sequencer State Diagram	25
Figure 10. Vehicle Data Channels	34
Figure 11. Vehicle Data Multiplexer	35
Figure 12. FIFO Timing	37
Figure 13. Vehicle Multiplexer Timing	40
Figure 14. PROM Format	43
Figure 15. Telemetry Receiver	47
Figure 16. Receiver Clocks	51
Figure 17. Hardware Control Loop	53
Figure 18. Demodulator Timing	57
Figure 19. Antenna and Cable Arrangement	59

ORIGINAL PAGE IS
OF POOR QUALITY

ACKNOWLEDGEMENT

The author would like to express thanks to all the members of the Mars Rover group that he has met during the past few years. Thanks go especially to Dr. S. Yerazunis for his unending motivation and inspiration, to J. Odenthal for his ideas and aid in debugging the design, and to Dr. D. G. Gisser and Dr. D. K. Frederick for their advice and guidance.

ABSTRACT

This paper describes a data link system used on a prototype autonomous roving vehicle. This system provides a means of acquiring, formatting and transmitting information on board the vehicle to a controlling computer. Included is a statement of requirements and the design philosophy. Additionally, interfacing with the rover systems is discussed, along with the overall performance of the telemetry link.

PART 1

INTRODUCTION

Since 1967 there has been research at RPI aimed towards an autonomous roving vehicle for the exploration of Mars. Due to the great distances involved, interplanetary exploration will have to be done by machines with local artificial intelligence.

The goal of this NASA-funded project at RPI has been to study and determine the feasibility of a hazard detection/obstacle avoidance system. The hardware centers upon a laser triangulation scheme where pulses of light are projected at the terrain ahead and the returns are electro-optically sensed and evaluated.

During the course of research, considerable hardware has been developed to support this computer vision system. A four-wheel drive roving vehicle, capable of negotiating uneven terrain, has been built and evolved over the years. Recently, capabilities have been expanded both on board the vehicle and off. Recently microprocessor control of the steering and propulsion system has been added to the vehicle. A second generation laser scanning system with greater resolution has been added. Additionally, a more powerful computer is now being used.

Due to the fact that the decision-making computer is not on board the vehicle, a telemetry data link is necessary in order to relay the real-time information to the host computer.

This paper focuses on such a system and describes its theory, operation, constraints and maintenance. The hardware de-

scribed has been built and tested and performance information is given.

PART 2
SYSTEM CONSTRAINTS

2.1 Data Sources

The telemetry system is designed to provide a communications link from the autonomous roving vehicle to the controlling computer. This vehicle has numerous sources of information on board.

The main source of information is the laser vision system. This unit, called the laser scanning mast, fires shots of infrared light at the terrain ahead, and encodes the returns it detects. Address information containing azimuth and elevation information of the laser shot as well as data information describing the detected return are gathered by the mast controller (see Craig's paper [1]). This data is buffered and appears as one of the sources of information to be sent to the computer.

The vehicle itself is the other major source of data. Since the rover must move under real-time control, information concerning its speed, pitch, roll, heading, steering angle, etc., must be included in this data-acquisition system. This data must also be buffered to allow for the access time involved in multiplexing many data sources.

2.2 System Requirements

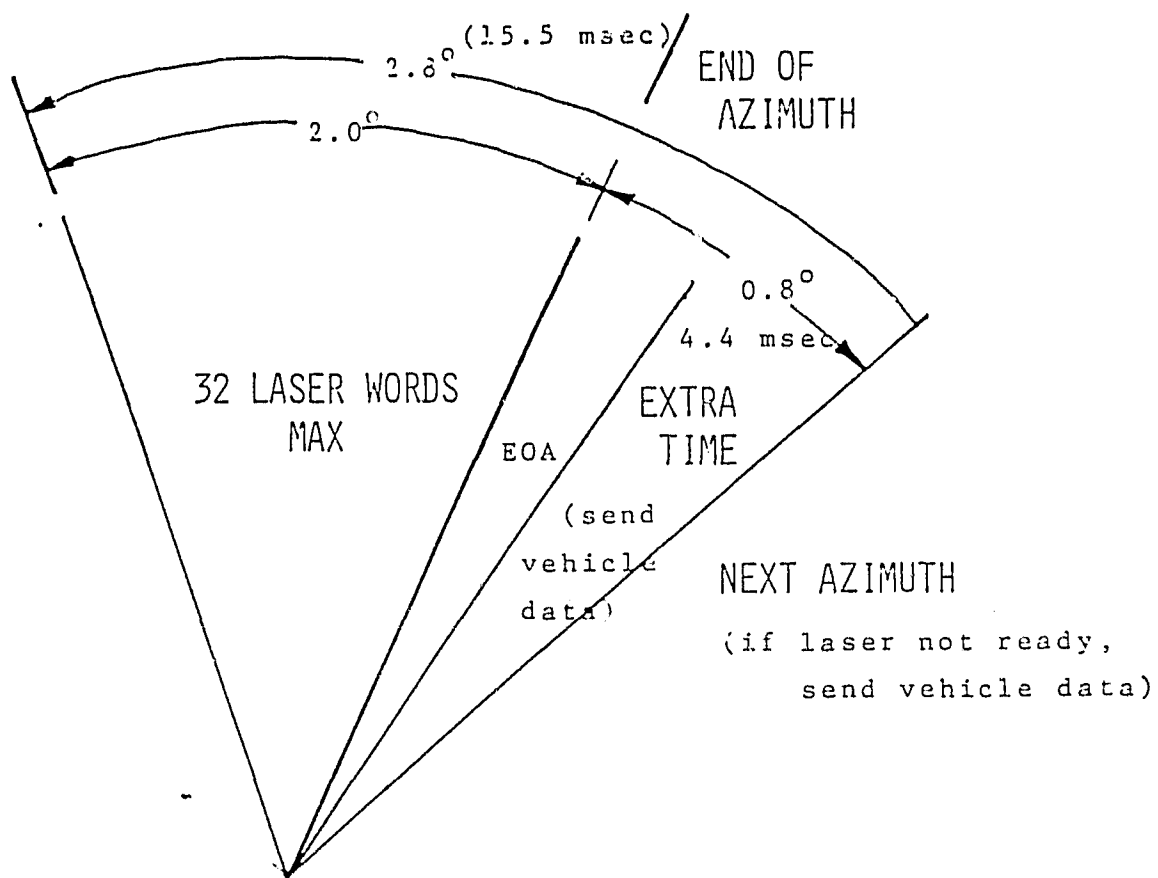
The main feature of the new telemetry system is its ability to pass data at the higher rate required to service the laser scanner. Additionally, the system puts the data into a 16-bit address and 16-bit data form, with each channel described by its own address. This allows future expansion capabilities and a full 16 bits of resolution if

necessary. The system also selects the format in which the output is transmitted so that the information can be quickly and efficiently processed. The format can be selected by programming the read only memory on the multiplexer board.

The speed constraint is set by the laser vision system. The laser mast can be programmed to fire laser shots in any pattern desired. The number of these shots determines the speed required. Taking a worst-case figure, there can be up to 32 laser shots per 2.8 degree azimuth, as shown in Fig. 1. The real-time computer software algorithms require a full block of vehicle information (gyro headings, wheel speeds, etc.), after each azimuth. This means that 32 laser words and then 32 vehicle words must be transmitted before the next azimuth (64 words per azimuth). This implies a rate of 242 microseconds per word, or a word rate of 4129 words per second minimum.

The telemetry frame format in Fig. 2 shows the serial frame. The ADLC (Advanced Data Link Controller Chip) generates the opening flag, error check bits and closing flag. Thus for a 64-bit-long word sent at 4130 words per second, we need a serial bit rate of at least 264.2 kHz. A bit rate of 500 kHz is used to allow for a margin of safety.

To allow for storage of data before it is sent, FIFO (First In, First Out memory) rate buffers are used in the laser electronics. These allow data to be shifted into storage as it is generated from the laser electronics and shifted out of the FIFO as it is transmitted. The use of FIFO rate buffers guarantees that no data will be missed when the transmitter is servicing another data source. Also, rate



$$15.5 \text{ mSec.} / 64 \text{ WORDS} = 242 \text{ } \mu\text{Sec.}/\text{WORD} = 4129 \text{ WORDS} / \text{Sec.}$$

$$\text{BIT RATE} = 264.2\text{kHz MINIMUM}$$

Figure 1. Laser Timing

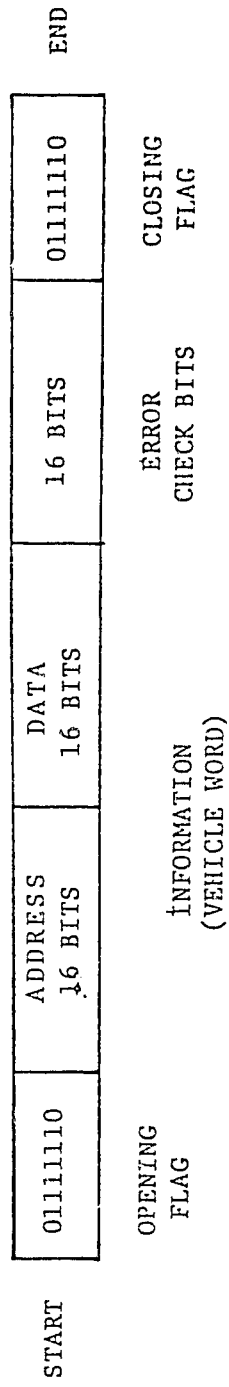


Figure 2. Telemetry Frame Format

buffering allows the information to be sent out in bursts or blocks.

The laser FIFO's are 40 words deep to allow for a whole 32 word azimuth block.

PART 3

TELEMETRY TRANSMITTER

The telemetry transmitter performs the functions of both deciding which data source to service and doing the actual serializing and encoding of data. The schematic diagrams are shown in the appendix.

3.1 Data Priority Network

3.1.1 Description. The data priority network is shown in Fig. TTL. Data input can come from either the four laser FIFOs or the four vehicle FIFOs. To minimize the number of interconnecting wires, the output enable lines ($\overline{CENABUS}$) on the FIFOs are used and the tri-state data lines from each source are tied together on one common bus. This bus contains 16 address lines (A_0 = least significant bit); 16 data lines (D_0 = least significant bit), and control lines for the two FIFOs. These lines are all sent over one 50-conductor ribbon cable to the telemetry transmitter card.

Due to the interrupt-driven nature of the real-time vehicle controlling software, the network must format the data it sends. Appendix 1 lists the transmitted interrupts. EOA (end of azimuth) and EOS (end of scan) generated in the laser controller and sent through the address FIFOs in bits A13 and A12, respectively, are described in detail in Craig's [1] thesis. EOA is generated after each set of azimuth shots and EOS tags the last shot of the last azimuth of a scan. This scheme allows the real time system to be notified that it has received a given azimuth set and the completion of a scan.

SCANNING MAST PLATFORM
(Electronic System Block Diagram)

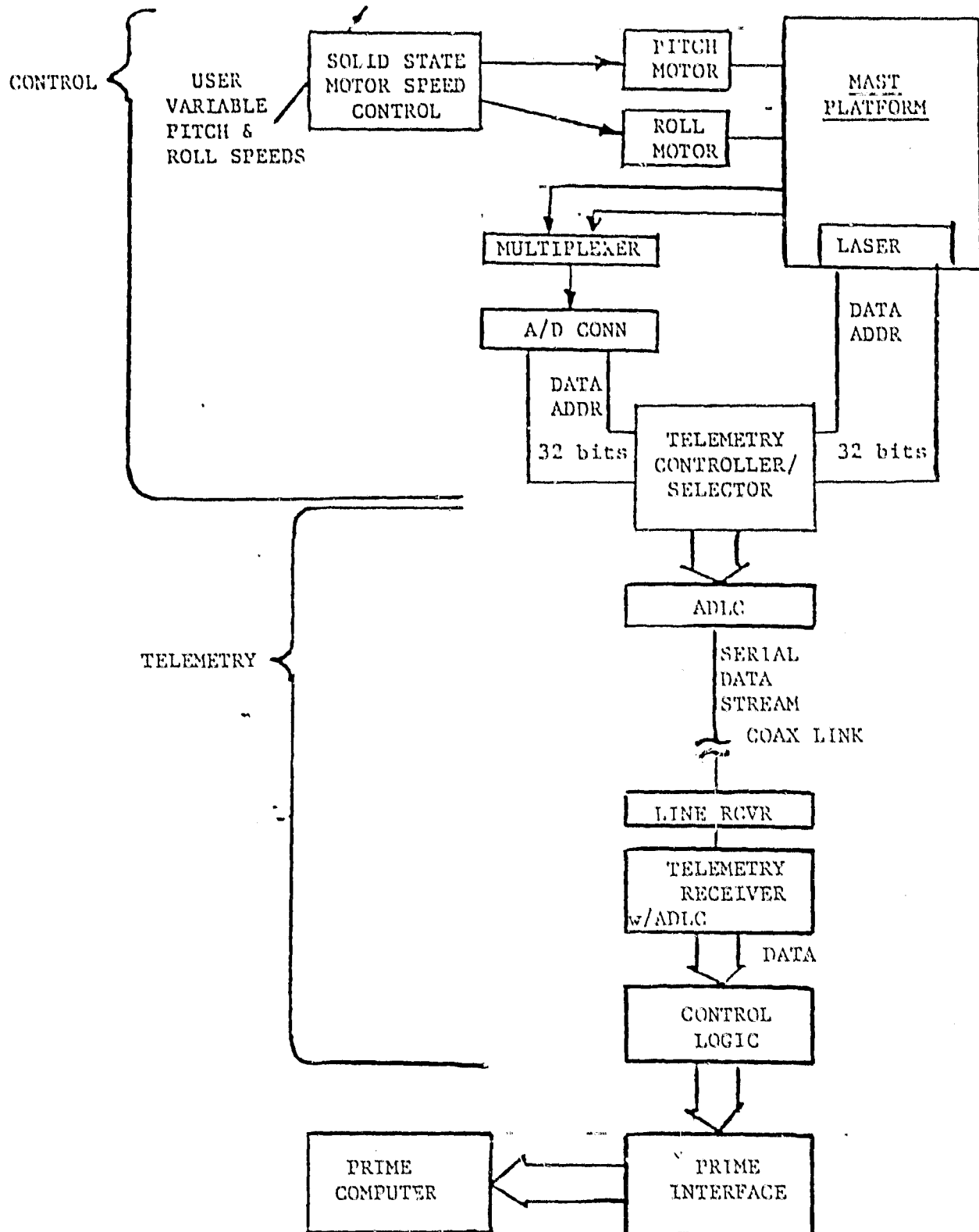


Figure 3. Overall Block Diagram

Another duty of the priority network is to assume that fresh vehicle data is sent along with each EOA and that vehicle data blocks are identified. Identification is accomplished by sending the vehicle multiplexer board EOY (end of vehicle) interrupt. This interrupt is placed on the most significant address bit of the last vehicle data word for a given block of data.

3.1.2 Design and Operation. The interrupt bits above are taken right from the bus and are used to set conditions in a J-K "mode" flip-flop (U21, schematic TT-1). After a system master reset pulse (MR) the "mode" (flip-flop, U21) gets cleared and the priority network starts in the "vehicle mode," with the vehicle data FIFO enabled and the laser FIFO output disabled. Clocks labeled with numbers are generated from the 1 MHz master clock with decoders (U39 and U40). Once in the vehicle mode, phase "2" shifts out the vehicle FIFO which causes the first vehicle word to drop to the bottom of the FIFO stack and OR VEHICLE (output ready) to go low after a short delay (see priority network timing diagram, Fig. 4). Next, phase "4" clocks the new conditions into the mode flip-flop (U21). The output ready flip-flop, U22A, will not be set if the FIFO that has been enabled does not have an output ready. This flip-flop is used to inhibit any change in mode or any new "shift in" pulses if the outputs were not ready. In this manner, if a data source is removed (e.g., disconnect laser mast for repairs), the data link will stay in the same mode and send only the available data. The conditions, determined by EOA, EOS and EOY are shown in the flow chart of Fig. 5.

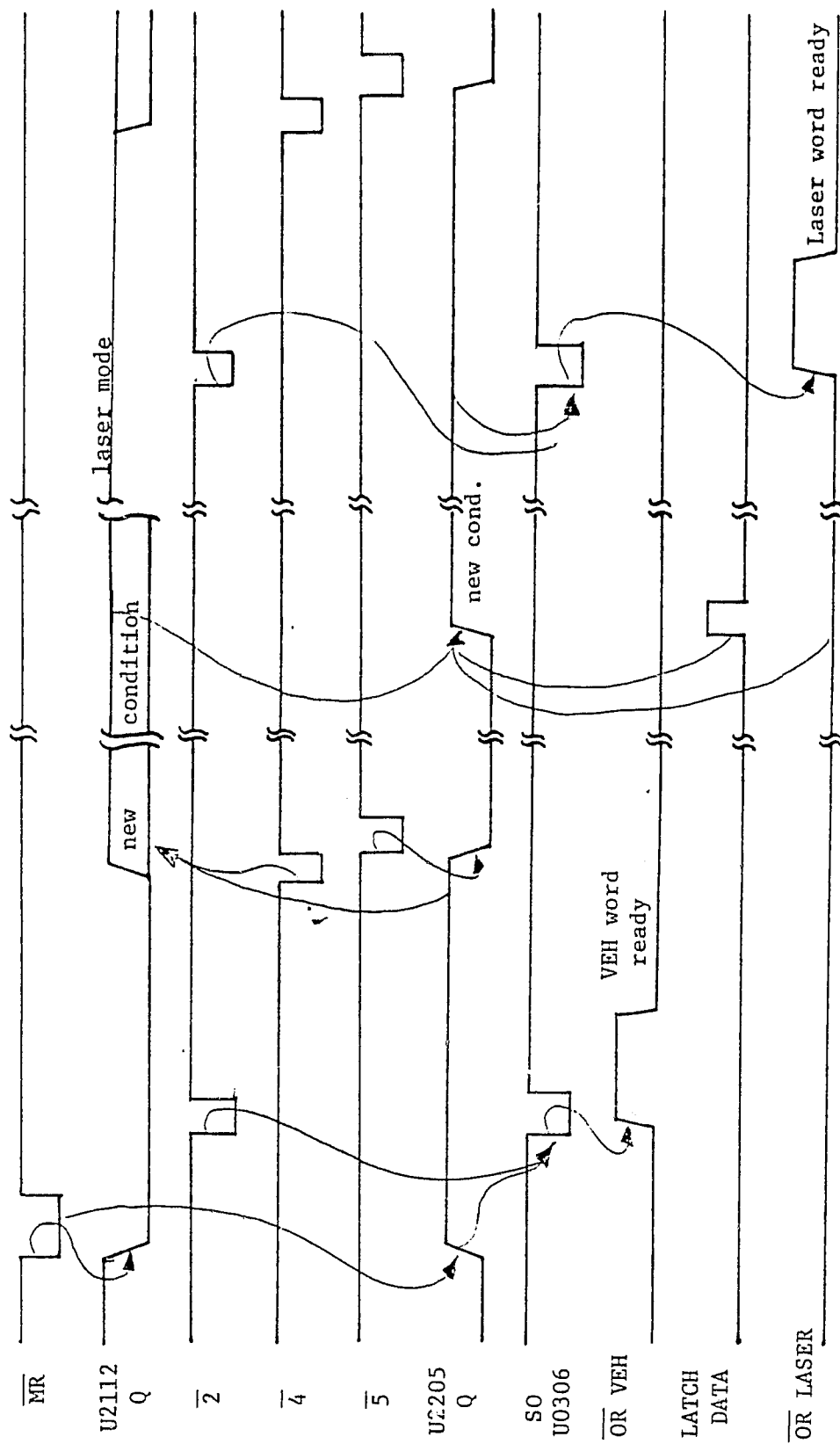


Figure 4. Priority Network Timing

Phase "5" now resets the output ready flip-flop so that a new test for output ready may be made. Latch data occurs next. This causes the complete 32-bit word to be stored in the telemetry latches for subsequent transmission. Also, "latch data" clocks the output ready flip-flop.

The cycle repeats, staying in the vehicle mode until an EOV is sent. Notice that the mode will not change until after the interrupt word has been sent, thus no words will be omitted in the data stream. Upon receipt of an EOV, the priority network switches into the laser mode until an EOA or EOS is sent. Either of these interrupts after laser data will cause the system to go back into the vehicle mode and send at least a block of vehicle data. Notice NAND gate U23C will only allow the EOV to change the mode if the laser is ready, otherwise another block of vehicle data will be sent.

A throughput calculation shows that, at worst, if an EOV occurs and just misses the laser's output being ready, the laser FIFO will not overflow while we are still in the vehicle mode. For a data rate of 242 sec per word, 32 words are sent in 7.26 msec. This means the laser FIFO, which takes 11.1 msec maximum to generate an azimuth, will not overflow the 40 word FIFO while filling it with an azimuth. Timing and a flow chart are shown in Figs. 4 and 5.

3.2 ADLC Operation

The heart of transmitter is a Motorola M68A54 advanced data link controller (ADLC). This chip is a complicated 6800 support device. It contains FIFOs and shift registers for rate buffering and parallel

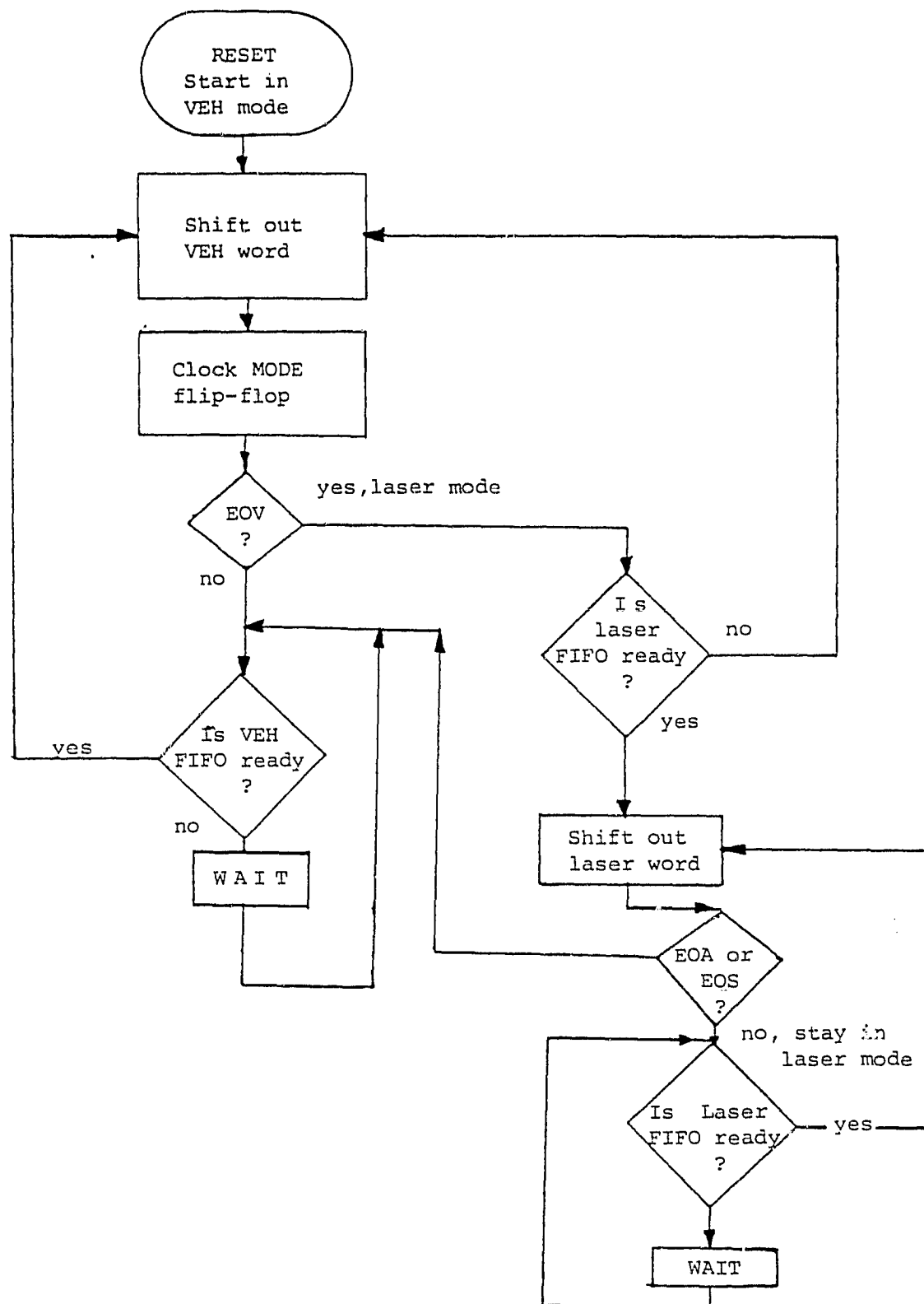


Figure 5. Priority Network Flow Chart

to serial conversion. It has control registers and status registers for a wide variety of functions. In this application, the device is used in a transmit data service request mode (TDSR). This is particularly well suited for the application as it is a DMA mode which requires a minimum of hardware support. The device was selected because of its high speed, variable length frame, low cost, excellent error detection qualities, low power consumption, and its compatibility with the M6800 microprocessor currently used on the roving vehicle. In fact, the design of the telemetry transmitter was simulated and debugged using the 6800 steering and propulsion microprocessor. To fully understand the operation of the M68A54, the reader is referred to the manufacturer's specification sheet (see the Appendix).²

The ADLC transmitter section has three bytes of rate-buffering (1 byte = 8 bits wide) which can be written to via its data port. Information put in this stack falls through to a parallel to serial shift register. The ADLC will take an information field and add to it a starting flag, error check bits and a closing flag. The frame format is shown in Fig. 2.

The error check sequence employs a cyclic redundancy error check character. The ADLC automatically calculates an error check character from the information field and this word is recalculated and compared at the receiver. This method of error checking is very reliable, and because of the high data rate, any word found in error may be neglected. In the case of 32 vehicle words, the data will not become stale if it is missed one time through the loop since it is

re-transmitted once every 7.26 msec. That is, any word received in error will not be changed in the computer, and the software can still use the previous value without any loss in real-time control capabilities since the real-time control loop is greater than 7.26 msec.

Internal to the controller chip, but accessible via the 8 bit bidirectional data bus are four control registers and two status registers. The device is configured during initialization. Like any 6800-type device, all the registers and buffers are memory-mapped and can be selected via the ADLC address lines. Data transfer is enabled by the chip select input and is transferred during falling edges of the E clock.

After the chip is reset, it must be initialized by writing the configuration desired into the four control registers.

TDSR is chosen because it is most efficient for direct memory transfer, which is being done at the receiver (computer interface). In the TDSR mode, when the transmitter's FIFOs request service i.e., they are empty) they signal this by setting their TDSR terminal high (logical 1), rather than signalling this through an interrupt request (IRQ). In this manner, time will not be wasted servicing an interrupt, but instead, TDSR will signal the controlling hardware so that the next byte can be sent.

Once the control registers are initialized, transmission may commence (specific control register details are given in the hardware description). Since the word to be sent (information field) is 32 bits long, and it can only be loaded into the ADLC one byte at

a time, there must be four one-byte write operations to the chip.

To signal the start of the frame, the first byte is written to the "frame continue" address. This will initiate the opening flag and start serial data flow at the bit rate selected by the transmit clock. Cued on TDSR, we can load the ADLC with the next two bytes of the word to be sent. To close the frame, we send the last byte by writing it to the "Frame Terminate" address. When the ADLC sees this, it includes the error check sequence and then the closing flag. This whole transmit operation is repeated continuously, sending the data appearing at the latches.

The operation of the ADLC in the receive mode is covered in Section 5.1.

3.3 Hardware Description

Now that the ADLC has been described, the support hardware can be discussed. Schematic diagrams TT-1 through 3 show the telemetry transmitter hardware. All transmitter system clocks are derived from a 1 MHz crystal oscillator (AV1). This clock is fed to two synchronous binary counters (AV2 and AV11) which are decoded by A39 and A40 to generate the various micro-phases, numbered 1 through 31. These counters sequence through until they reach a count of 35 when they are reset by gate A3B (schematic TT-2).

An initiating master reset can come from the microprocessor backplane bus or a pushbutton reset switch mounted on the transmitter board. The reset clears the ADLC and all counters and flip flops to assure a synchronous, deterministic start-up. Upon master reset, flip-

flop U45B starts in the INIT (initialization) mode with "Q" (U4509) low. This causes all bus drivers to the ADLC to be disabled and the open-collector initialization PROMs to be enabled. Firmware read only memories U43 and U44 contain the data and addresses necessary to write to the control registers and configure the ADLC. Thus, the transmitter can be made to run in any available mode simply by programming these PROMs.

3.3.1 Initialization PROM Coding. The present configuration for initializing the telemetry transmitter is shown in Appendix 2, along with the definitions of the control register bits. Since the ADLC is used as a transmitter here all control bits to the receiver are disabled.

The first PROM word sets up control register 1 (register select lines RS1, RS0, R/ \overline{W} equal 0, 0, 0). In this step we are put in the TDSR mode, all interrupt requests are disabled, the receiver section is reset, and the transmitter is temporarily reset. Next, to access control registers 3 and 4, register 1 is again addressed. This time address control (AC) is set and the transmitter reset bit is no longer set. AC allows the selection of CR3 and 4. Control register 3 (RS1, RS0, R/ \overline{W} = 0, 1, 0 with AC set) has all of its bits set to logical "zero" states, since we are not interested in any of these features except that of bit b3, which causes the transmitter to send a continuous string of ones (flag idle time fill).

We now select both opening and closing flags, as well as the byte length with control register 4. Bit b0 does the former, and

bits b1 and b2, both high, set the transmit word length at 8 bits. Of additional importance is the fact that b7 is made zero, which causes the serial bit stream to be in an NRZ format. NRZ means non-return to zero, that is, the serial data stays in the same state to send a binary "zero" and switches to the opposite state to send a binary "one."

The PROM sequence continues with an access to control register 1 in order to clear the AC bit. Next, register 2 can be selected where the system is put in a prioritized status mode to make it easier to read the status.

The next two words from the memory have system controlling functions. First, notice bit D3 of PROM 2 is no longer set so that the ADLC is not selected via "chip select" any more. In address "0110," a start pulse appears at bit D5 of the same PROM. This signal is used to take flip-flop U45 out of the INIT mode and start the sequencer counters (AU2, AU11). Notice that once the system leaves the initialization mode, this circuitry is no longer needed, hence the reason for gate U33A in TT-2. This gate inhibits any chip select pulses from the PROM once the system leaves the initialization mode. In addition, gate U33B guarantees that a START pulse will not be generated prior to initialization due to a race condition during master reset.

3.3.2 Transmitter Sequencer. After initialization, the sequencer depicted in schematic TT-2 is used to set up data to and control the operation of the ADLC.

Because the device has an eight-bit data port and thus can

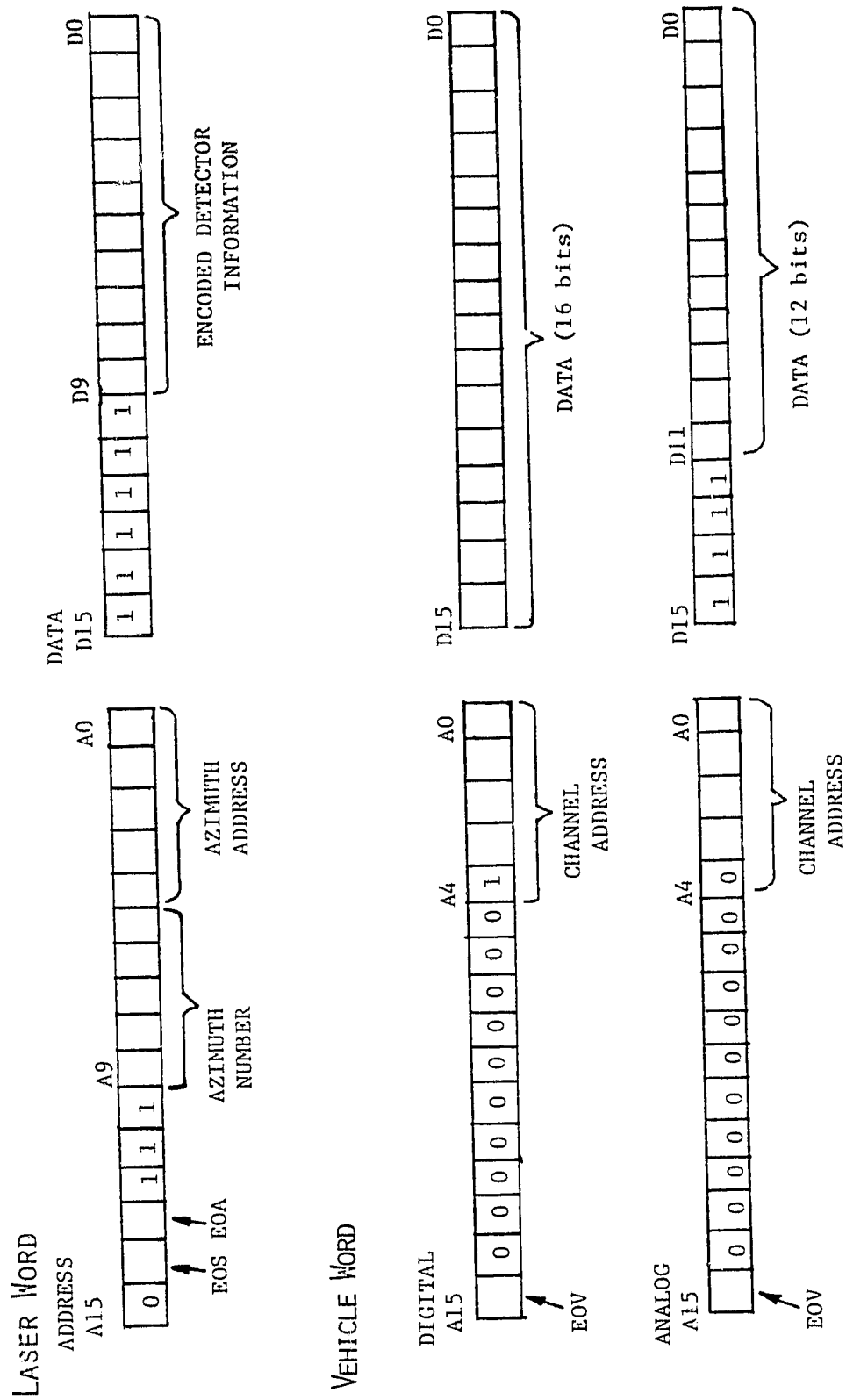


Figure 6. Word Format

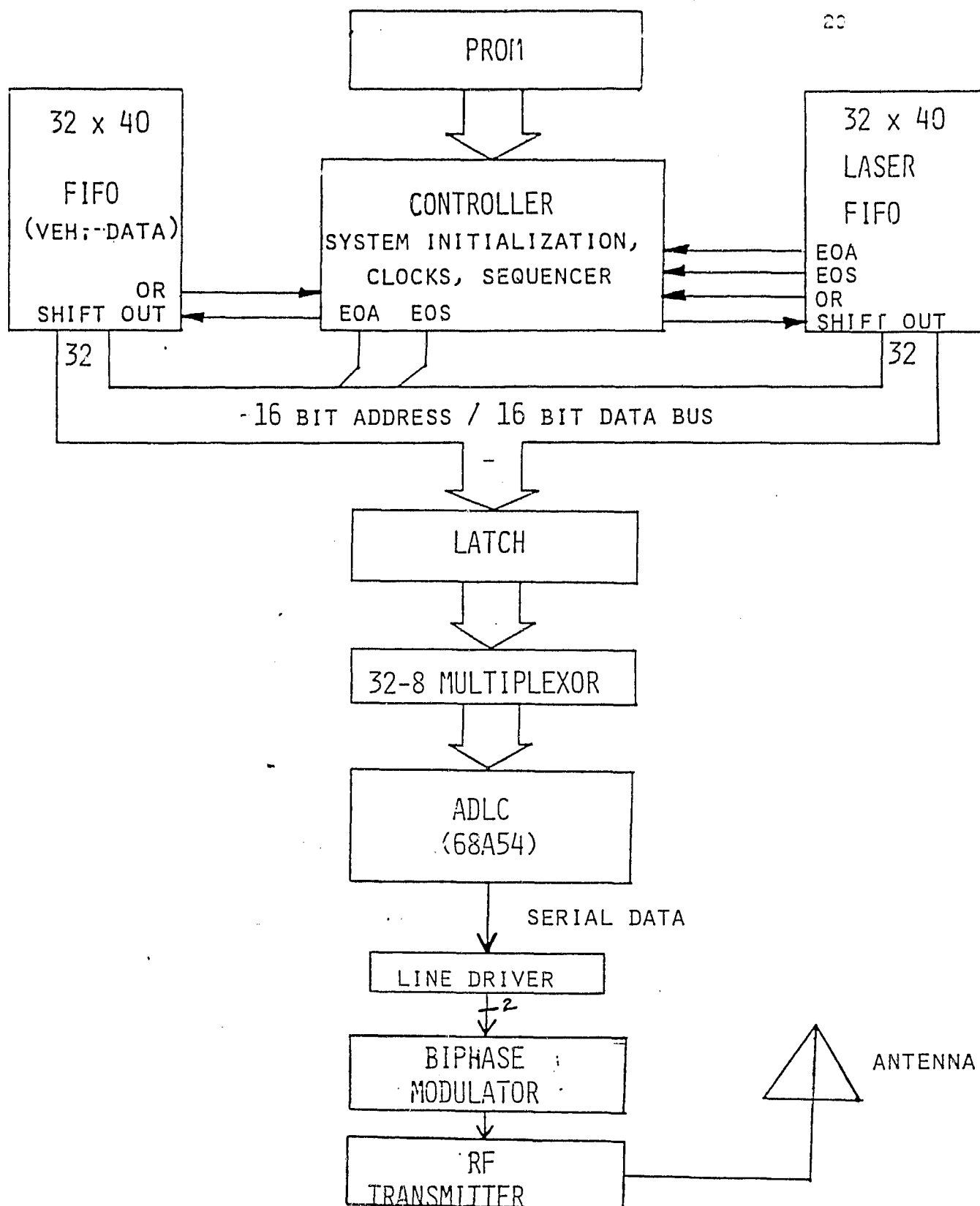


Figure 7. Telemetry Transmitter

only handle long words in blocks of eight, a means of chopping down the 16 bit address and 16 bit data word was employed. This is shown in the block diagram (Fig. 7) as the 32 to 8 bit multiplexer. Also a latch is used to hold the 32 bit word while the multiplexer is scanning and the FIFOs are getting the next word ready.

After clock phase "1" latches the 32 bits of data into the LS374s (U6-U9, schematic TT-3), the 32 to 8 bit multiplexer (U10 to 13) operates as follows. Counter U5 gets a BYTE SEL pulse (clock phase 2) and goes into the "00" state, selecting bits A0 through A7 in the multiplexers. The first byte of the word is now placed on the ADLC bus and phase "7" causes the "frame continue" address to be set up at driver U15. Notice at this point that the ADLC is clocked on ECLK ($\phi 2$) which is the invert of phase one ($\phi 1$). The opposite phase is used because all data and addresses are set up by clock $\phi 1$ and the read/write is done by the ADLC on $\phi 2$. The timing diagram in Fig. 8 shows this arrangement. This also illustrates one complete four-byte transfer and the synchronization with the FIFO shift out and priority network.

The loading of the first word into the ADLC causes its TDSR line to go low. This condition inhibits any further advance of counters AU2 and AU11 because TDSR tells when the next word can be loaded. The time it takes for TDSR to again return to a logical "1" state depends upon the transmitter clock rate, that is, how fast the serial data can be clocked out of the chip. This rate, of course, should never exceed the ECLK frequency. It is also important that all other operations

such as shifting out the laser and vehicle FIFOs will cease when the counters above are frozen.

When TDSR comes back up and the decoder (U39) reaches state "9," the byte select counter (U5) will be clocked in order to put the second byte (containing A8-A15) on the bus. When the sequencer reaches state "14," this new byte will be put into the ADLC at the frame continue address. After the ADLC loads this word, it again issues a TDSR and the process repeats. The cycle ends when the last byte (D8-D15) gets selected at the multiplexers (U10 to U13) during phase 23, followed by writing this byte to the ADLC at the "frame terminate" address (phase 28). This address ($RS0, RS1, R/\overline{W} = 1, 1, 0$) notifies the data link controller that the word to be read will be the last in that frame. Thus each frame contains a whole word, A0 through D15. The ADLC can now do the necessary housekeeping to send the word out in a serial stream.

The last thing the sequencer does is to load the byte select counter (U5) to its maximum count. This is done so that the first BYTESEL pulse of the next frame will start the counter at zero again, thus accessing the first byte. The sequencer runs continuously, shifting out data words from the FIFOs as it is ready for them. The repetitive cycle is shown in a timing diagram in Fig. 8, and a state diagram in Fig. 9.

3.4 Microprocessor Control

In order to provide an interface to the system with a microprocessor, there is some additional circuitry on schematic TT-3. Com-

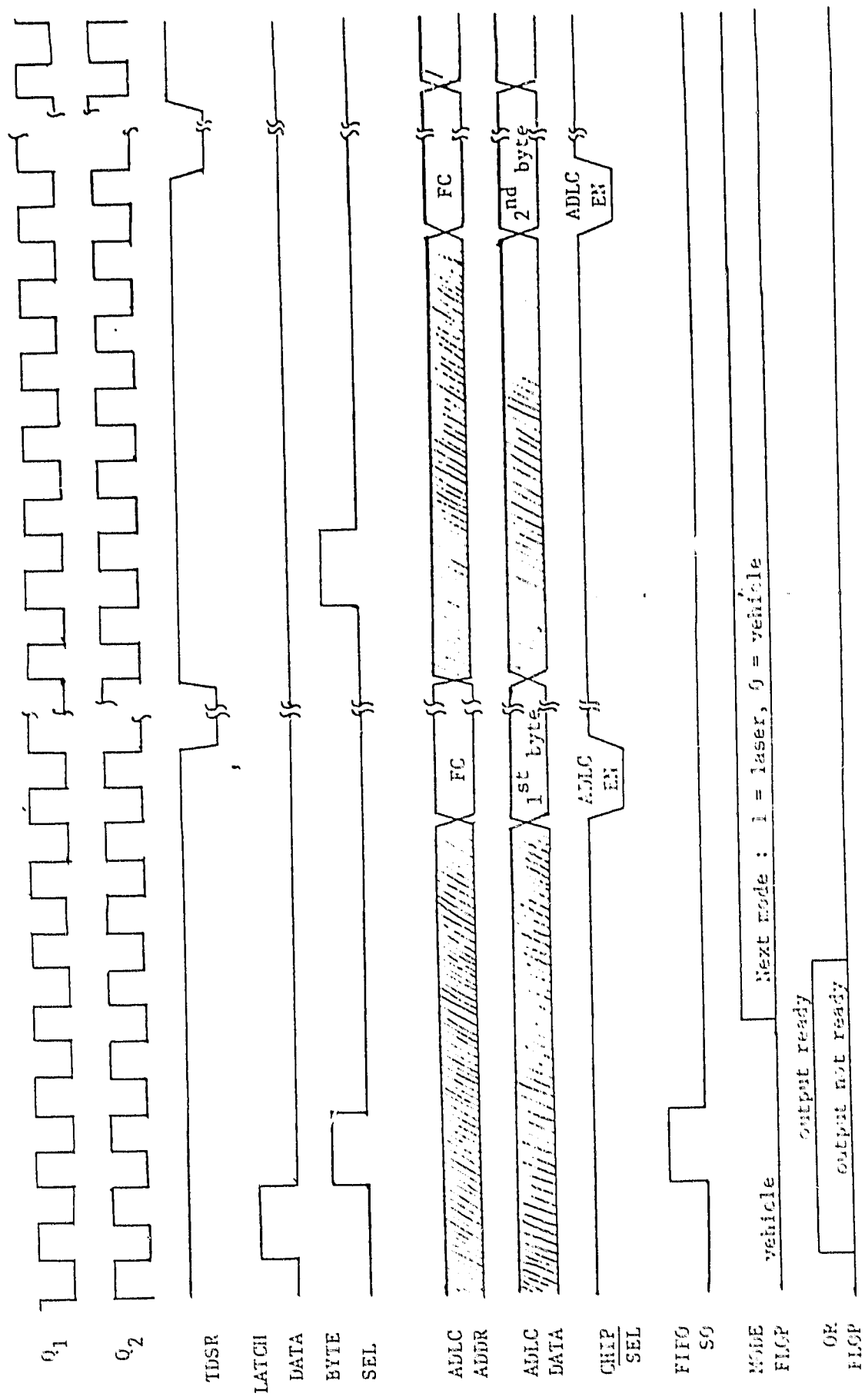


Figure 8a. Telemetry Transmitter Timing

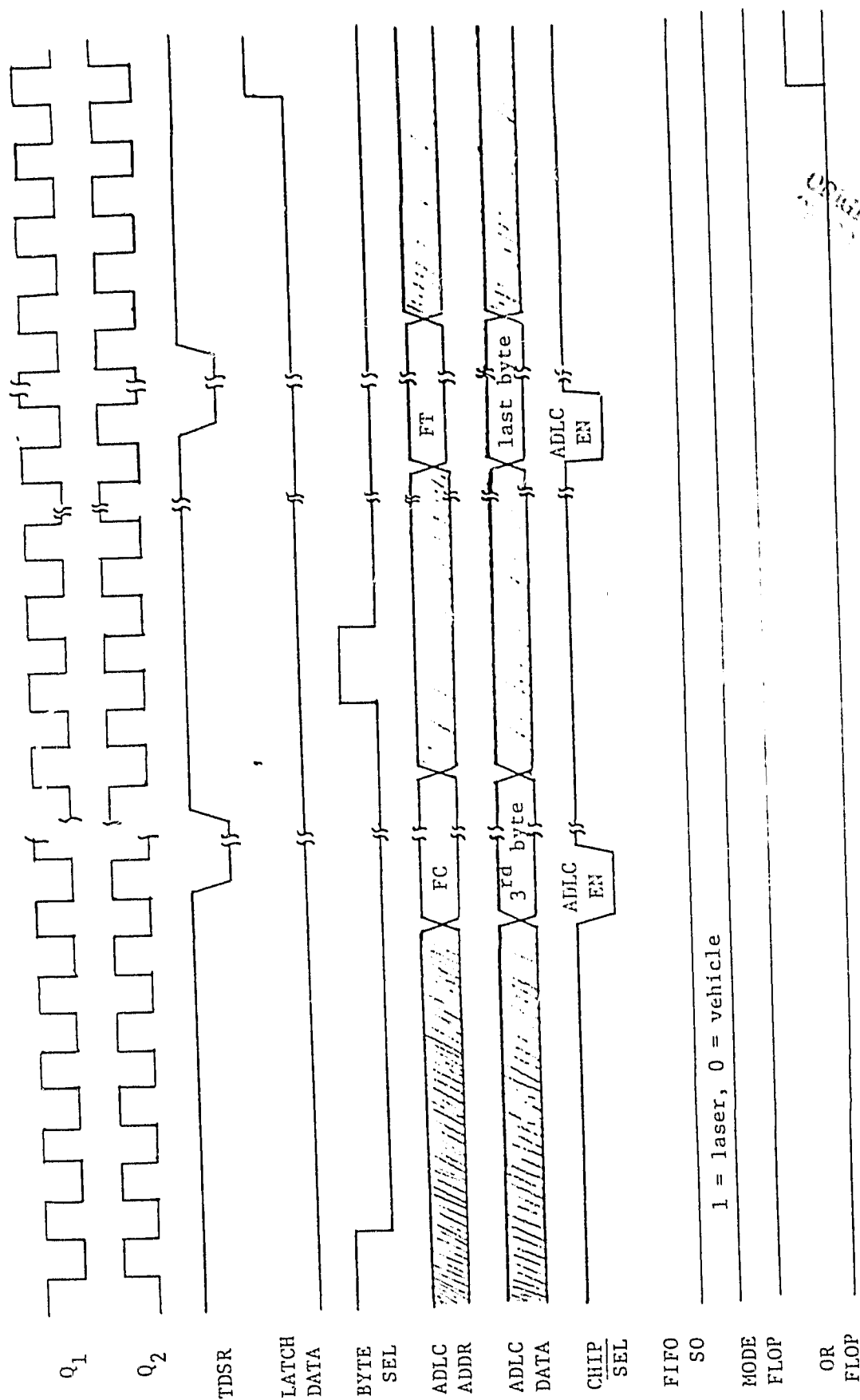


Figure 8b. Telemetry Transmitter Timing

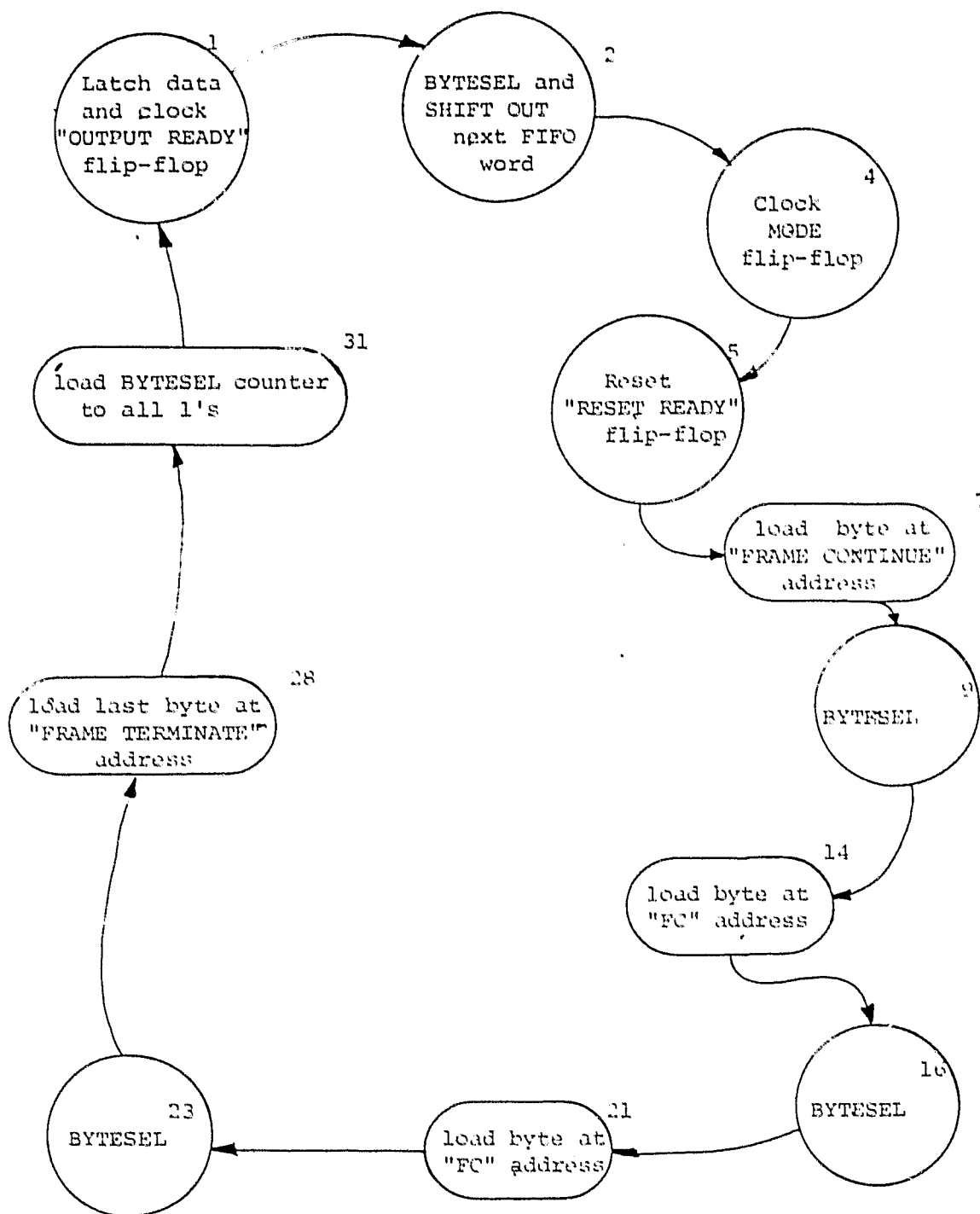


Figure 9. Telemetry Sequencer State Diagram

paritors U29-U32 check the microprocessor address bus with respect to the 14 dip switches of SWA and SWB. When these addresses match up, we will get an equal condition on U32 and the address bus can be enabled so that the ADLC will get lines RS0, RS1 and $R\overline{W}$ put under the control of the microprocessor bits A0, A1 and $R\overline{W}$, either the bus receivers (U17, U18) or transmitters (U36, U37) will be enabled.

Direct control by the microprocessor can be accomplished by writing to the correct spot in memory. Gates U34 do the address decoding of A3 and A2 to provide the start and stop functions. For example, when the top 12 address bits match the dip switches, then the bottom (least significant) bit value of "1XXX" would generate a "start" pulse and the value of "X1XX" would generate a "stop" pulse. These signals were used to go from the INTR mode to the INX mode at flip-flop U46B, schematic TT-2. It should be noted, however, that since the telemetry transmitter presently runs without the need of the microprocessor, the signals from the address decoders above are disconnected. If any further evaluation with the 6800 is desired, these can easily be reconnected.

3.4.1 Software. During the course of the evolution of the telemetry transmitter, a microprocessor routine was used to simulate the necessary hardware. A flow chart, M6800 coding and equivalent addresses used to signal the transmitter hardware are shown in Appendix 3.

which of the four bytes of the 32 bit word are to be loaded into the ADLC. RYTESEL is clocked to signify the second word and the status register is read repetitively until bit "0" of status register 2 is set, meaning the ADLC is ready for the next word. When this occurs, the status is cleared, the multiplexer is tied to the ADLC data port and the byte is loaded at the frame continue address.

This repeats until accumulator B reaches 4, when the byte counter is to select the last byte. We again return the bus to the multiplexer and enter the last byte at the frame terminate address. The byte select counter and accumulator "B" are initialized and the process repeats by returning to "C" in the flow chart.

The speed at which the transmit clock can be run is limited by the time it takes for this program to acknowledge the presence of the TDRA status bit.

3.5 RF Transmitter

The preceding hardware provides an NRZ serial bit stream with clock. If the system is used on the dynamic test platform within the lab, a wireless link is not necessary, so the data and clock may be sent directly to the receiver on the general purpose interface board (GPB) via the two 50 ohm coaxial cables provided. The receiver has a 50 ohm input impedance and the line drivers (U48A and B, schematic TT-3) on the transmitter yield sufficient current source and sink capabilities with the pull-up pull-down arrangement in the receiver.

For use on board the autonomous vehicle, the data must modulate a radio frequency carrier. One problem with RF transmission is that

some means for sending along clock information must be included in order to use a synchronous ADLC at the receiving end. After surveying several encoding and modulation techniques, it was decided that a biphase modulation technique would be simplest and easiest to implement with the present bandwidth requirements and receiver available. Biphase format was chosen because it provides a data transition every clock period. Thus, when received, these edges can be pulled out and a phase locked loop can be used to recover the clock.

The actual RF-carrier modulation used is pulsed amplitude modulation (PAM). PAM has the advantages of being simple to implement and easy to demodulate at the receiver. The bandwidth required is at least twice the modulation or transmit clock rate. Thus for a 500 kHz transmit clock, a minimum RF bandwidth of 1 MHz is necessary. This is provided at the receiving end through the use of a wide bandwidth television-type front end. A "Quasar" tuner and IF strip are used, providing an IF bandwidth of about 4 MHz.

If the rover is to be tested locally, say within one mile of the receiver, an RF power level of one watt was found to be satisfactory for reliable communications (see Section 6.1). The previous data transmitter ran about 80 milliwatts and seemed to be unsatisfactory for reliable communications over the specified distance. A power amplifier was built for the first transmitter and some modifications were made to increase the power output of the original transmitter to about 600 mw. This first transmitter was at a frequency of about

202 MHz, or lies in the middle of the commercial television band. With the higher power level, it was found that the telemetry transmitter interfered with the command receiver on board the vehicle, since the command receiver operates at a frequency of about 156 MHz.

Instead of going the route of isolating and filtering the transmitter and receiver on the vehicle, which would take much time and suitable equipment, it was instead decided to build another telemetry transmitter which would operate on a frequency far enough removed from the command receiver on board the vehicle.

In order to use the same telemetry receiver back at the JEC (the present receiver is very satisfactory in terms of sensitivity and bandwidth), a frequency of 53.9 MHz was chosen which lies just below the ch. 2 TV band. This frequency is within the tuning range of the present receiver and far enough from the command receiver to eliminate interference. Also, a more reliable signal will be received from the rover due to the longer wavelength.

3.5.1 Hardware Description. The telemetry data transmitter consists of four stages. These are the oscillator, driver, power amplifier and modulator. Referring to the schematic, the first stage is the oscillator. This stage generates the fundamental 53.9 MHz signal. For frequency stability and reliability, a commercial CE ICOM (internally compensated oscillator module) was used. This module will operate from a supply voltage of 5-15 volts, and is being used with a supply of 12 volts. Options are provided for frequency modulating the oscillator and for keying the oscillator on and off. FM is not

employed so this terminal is tied to a fixed 1.0 volts. The power amplifier is run in a class C mode where with no input it will draw no idling current and thus keep power consumption down to a minimum.

The oscillator signal is taken off the collector of Q_1 at point 2 and fed through a parallel tuned circuit. L_1 was calculated to be 1 uh to resonate with C_1 , a miniature ceramic variable capacitor. All inductors are wound on 3/16 inch diameter slug tuned forms. These are variable and compact.

The driver transistor Q_2 is biased in its linear operating region by voltage divider R_1 and R_3 . Signals are fed to the base of Q_2 by a secondary link on L_1 . This was found to be most effective both for impedance matching and maximum transfer without loading down the oscillator stage. The driver stage is modulated by keying the voltage to R_3 by a totem-pole transistor arrangement with the modulating digital signal.

The capacitors in the keying circuit and capacitor C_3 band-limit the modulating signal. RFC_5 isolates the RF from the modulator.

For greater stability and to limit the current through Q_2 , an emitter resistor and bypass capacitor R_4 and C_4 are used. Collector current for Q_2 is supplied through an RF choke and a current limiting resistor of 510 ohms. Thus total collector current will be 12/610 or 19.6 mA which is within the maximum current from spec sheets for Q_2 .

The power amplifier stage is DC biased through RFC_3 to run class C. Transistor Q_3 is rated for better than 20 dB of gain at 50 MHz. Thus 100 mW of drive should yield over 1 watt output. Output is taken off of an L-C tuned circuit. RFC_4 supplies collector current

for Q3. Tests have shown that for 1.5 watts DC input power to the final stage (I_c of 125 mA) about 1 watt RF output was achieved. This is approximately 66% efficiency.

All the supplies are bypassed with capacitors of suitable reactance at 50 MHz to provide good isolation.

A 50 ohm antenna is used with this transmitter. The antenna is a $1/4$ wavelength whip with a base loading coil. This provides omnidirectional radiation pattern so the vehicle may be pointed in any direction with respect to the receiving antenna with no difference performance.

The transmitter schematic is shown in Fig. TT-6.

3.6 Physical Layout

The telemetry transmitter has been built on a Motorola M6800 family wire wrap circuit card. This is compatible with the microprocessor used for propulsion and steering, and fits in the same card cage. The five volt supply and microprocessor bus are connected via this 86 pin EXORCISER card edge connector.

The signals from the laser and vehicle FIFOs are brought in on a 50 pin ribbon cable to a connector on the card. The transmitter data and clock exit the card via the BNC connectors labeled TXD and TX CLK, respectively. Additionally, the MHz crystal is in a socket adjacent to these connectors. Schematics TT-4 and -5 show the board layout from each side.

Wire-wrap sockets were used for ease in modifications and increased flexibility. The power and ground lines are distributed ver-

tically, in a matrix-arrangement to decrease ground impedance and two decoupling capacitors are used per vertical row. Also all critical high-speed clock lines on the board are wired using twisted pair wire-wrap wire for minimum cross-talk. Conventions for the wiring are white wire for grounds, red wire for plus five volt supplies and blue and yellow for point to point wiring.

Technologies used are largely TTL and LS TTL, along with the MCS ADLC chip.

When used on the vehicle, the BNC connectors are wired to the RF box via 50 ohm (RG 58 A/U) coaxial cables with appropriate connectors.

The telemetry board is operational to date and the transmit data rate has been strapped for a 500 KHz TX clock. Other rates may be chosen by selecting appropriate dividers or by providing an outside source.

The transmitter circuit card is shown in Appendix 7.1.

PART 4

VEHICLE MULTIPLEXER

Part of the function of the telemetry is to provide a multi-channel data acquisition system for the information on board the roving vehicle. Presently there are 14 data channels on the vehicle, 11 analog and 3 digital. The system however provides up to 32 channels, or more, with some slight hardware changes. Fig. 10 shows the present vehicle data channels.

The basic philosophy behind the vehicle multiplexer board is to arrange the data in a given order and put it in a FIFO to be sent out as needed by the transmitter. This is accomplished by time-division multiplexing the channels in an order described by a programmable read-only memory (EPROM). A block diagram is shown in Fig. 11.

Since the computer processes digital data, any analog data on the vehicle must be converted into the corresponding digital format. We require many possible channels, but do not intend to send all data continuously, so an analog multiplexer may be employed to select the desired channel. This will allow us to use only one A/D converter to save cost.

Another feature of this system is its ability to service a shared RAM (random access memory) located on the propulsion and steering circuit board (see Bodgan's paper [3]). Specifically, data from the vehicle may be written to the RAM and certain data in the RAM may be sent via telemetry.

4.1 Hardware Description

Starting with the analog inputs, each has its own gain poten-

<u>CHANNEL</u>	<u>ADDRESS</u>	<u>INFORMATION</u>
0	00000	NOT USED
1	00001	" "
2	00010	LEFT REAR TACH
3	00011	RIGHT REAR TACH
4	00100	PLATFORM PITCH
5	00101	PLATFORM ROLL
6	00110	LEFT TORSION BAR
7	00111	RIGHT TORSION BAR
8	01000	FRONT AXLE ROLL
9	01001	GYRO PITCH
10	01010	GYRO ROLL
11	01011	CENTER TORSION BAR
12	01100	RIGHT FRONT TACH
13	01101	SELECTED WHEEL CURRENT
14	01110	LEFT FRONT TACH
15	01111	GYRO DIRECTIONAL
16	10000	STEERING ENCODER
17	10001	LASER CENTER SCAN
18	10010	COMMAND LINK ECHO
19	10011	NOT USED
20	10100	" "
21	10101	" "
22	10110	" "
23	10111	" "
24	11000	" "
25	11001	" "
26	11010	" "
27	11011	" "
28	11100	" "
29	11101	" "
30	11110	" "
31	11111	" "

Figure 10. Vehicle Data Channels

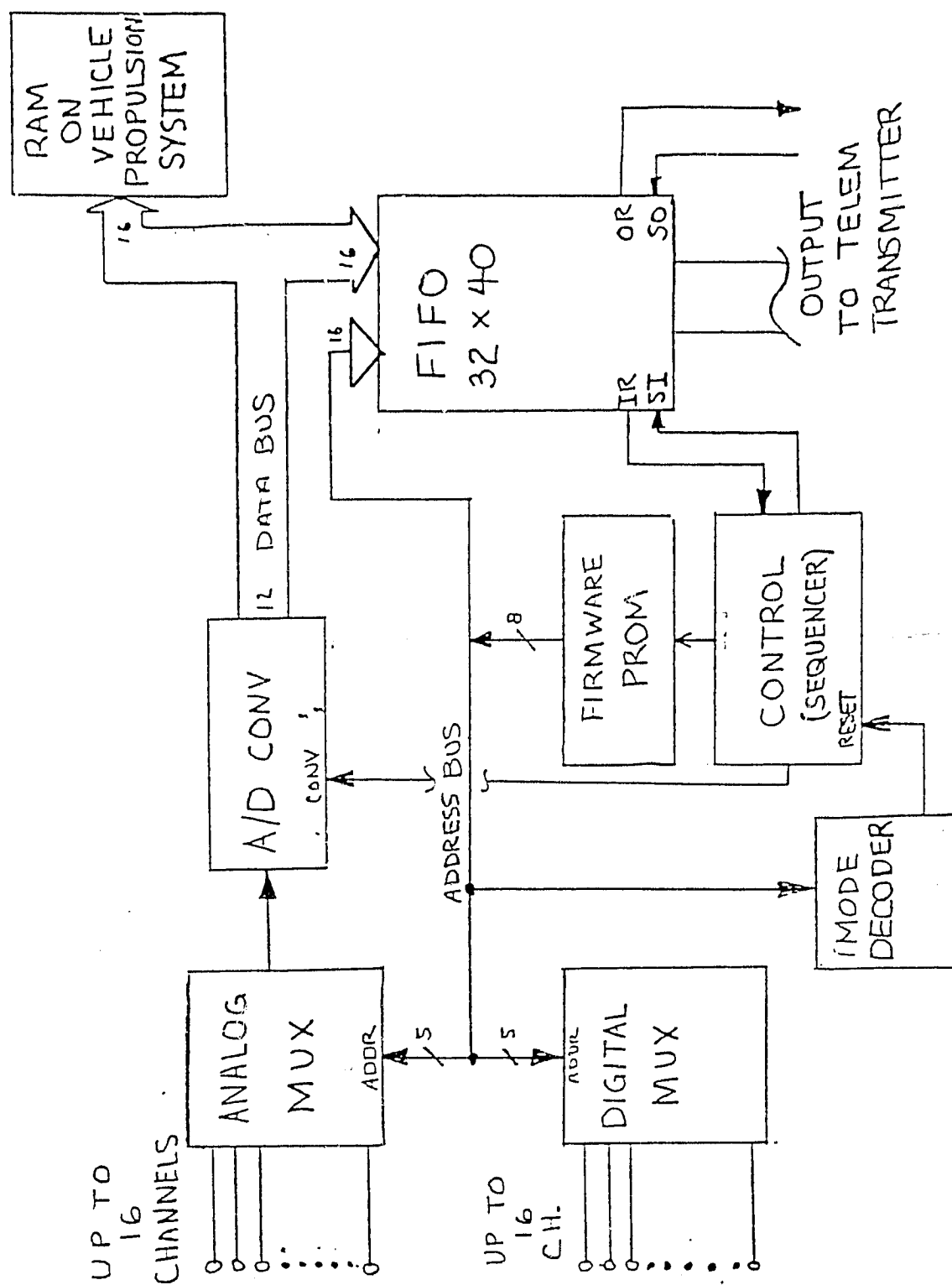


Figure 11. Vehicle Data Multiplexer

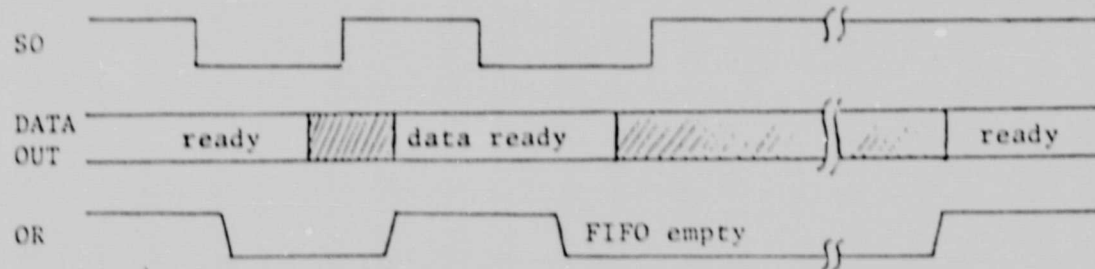
meter to set the scale factor desired. The analog multiplexer used is an Analog Devices AD7506. This provides for 16 possible channels, selected by a four-bit TTL signal. The "on" resistance is typically 300 ohms, which can be nulled out by the gain setting potentiometers.

The heart of the analog section is the analog to digital (A/D) converter. The part used is an Analog Devices ADC-12QZ, with 12 bits of resolution and a 40 μ sec conversion time. A 1458 input buffer is used before the A/D to make this point look like a high impedance. Because of this, the loading on the wipers of the potentiometers is negligible, and the inputs look like 20k ohms for any setting of the gain potentiometers. Offset nulls are adjusted in the A/D by the potentiometer labeled "zero adjust."

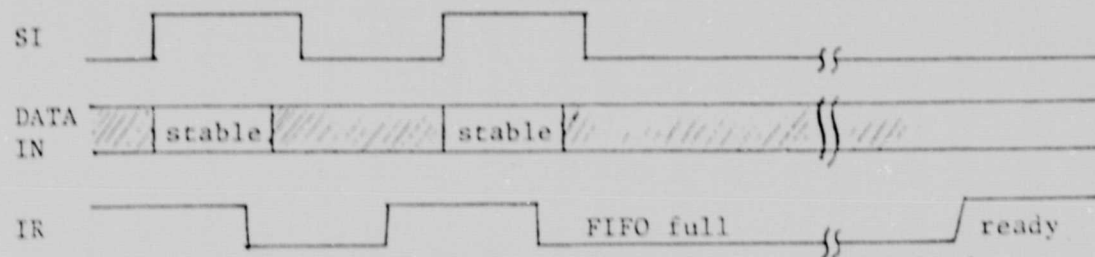
The circuitry in drawing TT-8 is used to sequence through the data channels and fill the FIFOs. So that complete sets of data may be pulled out of the FIFO, it is loaded in as blocks.

4.1.1 FIFO Rate Buffer. The first in-first out memories used on the vehicle multiplexer board are four Fairchild 3351 devices [4]. These are used in parallel to form a 32 bit wide word. In this configuration all four are shifted in simultaneously to clock the wide word.

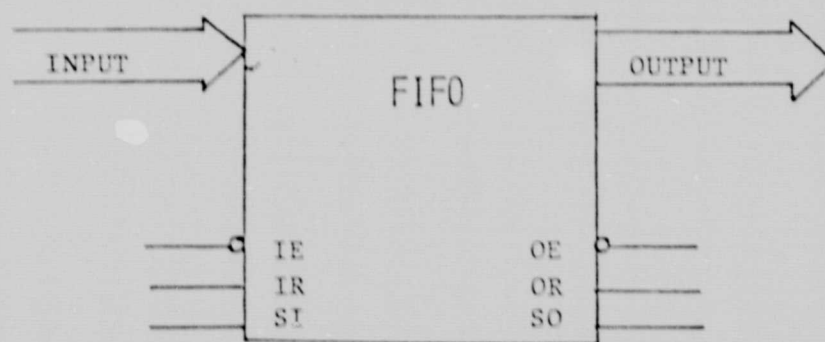
On the input the FIFO has "shift in" and "input ready" signals. "Shift in" is used to enter data into the FIFO, while "input ready" tells when the chip has empty locations, that is, when we can shift in data. At the output side of the device, there are "shift out" and "output ready" signal lines. When a rising edge is applied to shift out, and there is data in the FIFO, output ready will rise, along with



a. Output Timing



b. Input Timing



c. Symbol

Figure 12. FIFO Timing

data falling to the output lines at the bottom of the stack after a short delay. When shift out falls, output ready will fall after a delay. If shift out again rises, output ready will not come back up until new data is ready (see Fig. 12a). It is important to remember that the data remains on the output port of the FIFO until a rising edge on shift out clocks this word out and the next one falls into its place.

In order to cascade these devices in parallel the conditions where all the input readys are high and all the input ready lines are low must both be sensed. New data can be clocked in only when all the input ready lines are high.

4.1.2 Sequencer. It was desired to send blocks of vehicle data to telemetry while not preventing writes to the RAM during the laser time. This was accomplished by creating a "FIFO mode" during which the timing is interrupted by IR, and a "RAM mode" when data is written to the shared RAM regardless of the FIFO's condition.

It is a simple matter to sense when EOVI (end of vehicle interrupt word) is shifted into the FIFOs. It is more difficult, however, to sense when the FIFOs are empty. Monitoring OR (output ready) is not enough since OR goes low every time \overline{SO} (shiftout) occurs. Schematic TT-8 shows arrangement to indicate when the FIFO is empty. Flip-flop 19B ensures that OR has been low for at least 12 μ sec before \overline{COUNT} drives the system into the FIFO mode. This value was chosen because the worst-case bubble-through time from the FIFO spec sheet is 9 μ sec.

Flip-flop 19B goes high if the FIFOs are empty when 2^3 goes

low. If \overline{OR} has not gone low before \overline{COUNT} goes low, the FIFO is presumed to be empty and flip-flop 19A indicates the FIFO mode. The EOV input on U19A allows exit from the "FIFO mode" after the last vehicle word of the block has been shifted into the FIFOs. In the RAM mode the \overline{IE} of the FIFOs is held low, disabling any further shift-ins. Furthermore, during the RAM mode the feedback from IR, which in the FIFO mode halts timing until IR is ready, is held low allowing continuous "writes" to the RAM, regardless of the condition of the FIFOs.

The other important part of this circuitry is the network which "stretches" the shift in pulses. This circuit prevents a shift in unless all the FIFOs are ready, and stretches the shift-in pulse until all FIFOs complete their shift in. Phase \overline{Q} of flip-flop U25A will go high only if gate U7A shows that all FIFOs are ready. The shift in remains high until gate U22A shows that all the inputs are no longer ready, indicating a complete shift-in.

The clocks, shown in schematic TT-9, are all generated from the microprocessor phase one. This was done because this signal is a convenient, stable clock, and it will allow synchronous transactions with the shared RAM. Counters U18A and B divide the 921 kHz clock, and decoder U13 picks out the desired signals. A timing diagram for this arrangement is shown in Fig. 13. The falling edge of \overline{COUNT} and the rising edge of $\overline{CONVERT}$ are two microseconds apart in order to give the analog multiplexer (U12) sufficient time to select the proper channel. This is particularly important as this chip is CMOS and the analog signal must be correct at the A/D converter when it gets a $\overline{CONVERT}$ signal.

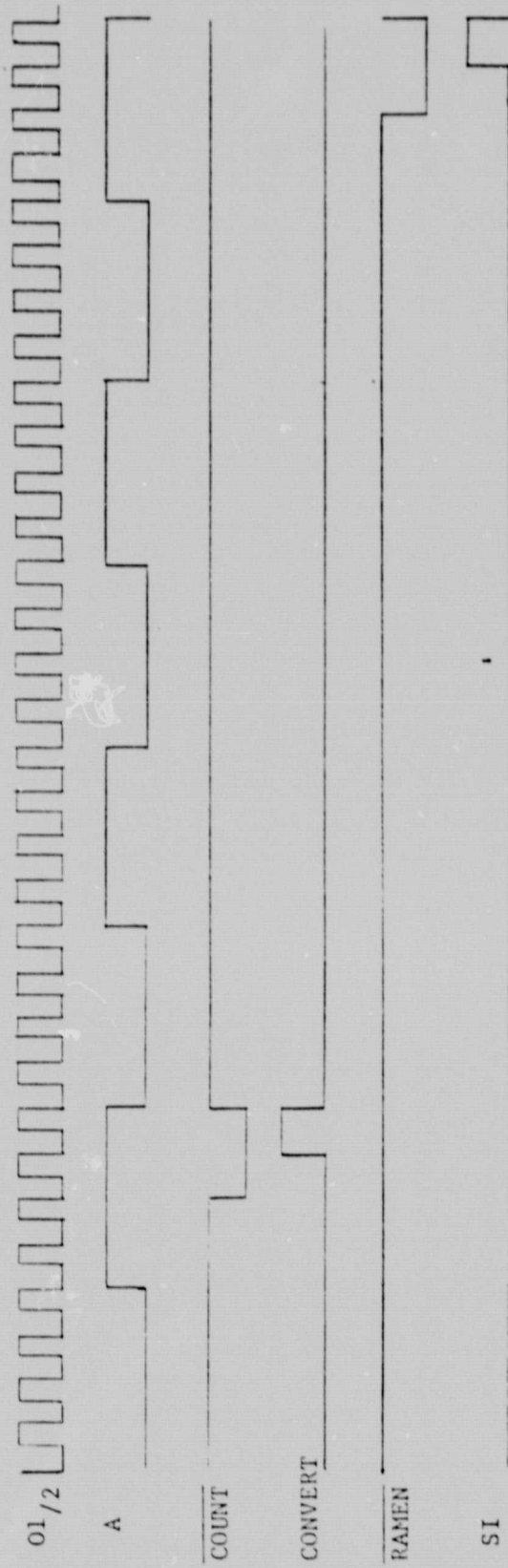


Figure 13. Vehicle Multiplexer Timing

The SIA signal, which causes a shift in to the FIFOs via flip-flop U25A, occurs 48 microseconds after the CONVERT pulse. This allows sufficient time for the A/D to make a conversion, since the part is rated at 40 microseconds to do a conversion.

Each time a word is shifted-in in the FIFO mode, gate U20B resets the sequencer U18.

Counter U1 on schematic TT-8 is used to clock the PROM which selects the data channel. This counter is clocked at falling edge of the COUNT pulse and reset after each block of data (when an EOY occurs).

4.2 PROM Format

The PROM used is an M68708. This PROM is eight bits wide and contains 1024 words. This device was chosen because it is readily available (equivalent to a 2708) and is erasable. Also, a programmer was built which interfaces to the propulsion and steering microprocessor. Thus new patterns can be readily programmed within the lab.

Data from the PROM directly addresses the muxes. The lower five bits (D_0 through D_4) select the channel number. These five bits are also used directly by the FIFOs for the address of the word sent. These 32 possible words are more than sufficient for the present roving vehicle. If more channels are needed, addition decoders and muxes may be added.

Bit D_4 from the PROM is used to differentiate between an analog or a digital word. If bit D_4 is set (logical 1), it will enable the 1 of 16 docoder, U5 (schematic TT-8), and disables bus drivers U16 and U17. Thus all digital channels have their own three-state drivers

which are enabled by the decoder U5. This bus is tied to the inputs of the FIFO.

When bit D_4 is low (analog word) drivers from the A/D converter (U16 and U17) are enabled, allowing only the A/D to deposit its 12-bit word into the FIFOs. Note that in the analog mode we have only 12 bits of resolution while the top four bits float high. This, however, is not a problem as the real-time software neglects these top bits for an analog word.

Additional information is contained in the PROM. The top three most significant bits (D_7-D_5) drive a decoder. Up to eight functions can be selected from this decoder, and three are used. A seven, corresponding to $[D_7, D_6, D_5 = 1, 1, 1]$ will tag the present word so that the EOVI interrupt bit will be set. This condition is fed into the FIFO in the top address position with the present word. This condition should be programmed in with the last word in a block to signal an end of vehicle data interrupt. Likewise, for simulation purposes, the laser data source may be simulated by sending an EOS (end of scan) interrupt corresponding to a bit pattern of 6 $[D_7, D_6, D_5 = 1, 1, 0]$.

If decoder U4 receives a "1" then counter U3 will be reset to its initial zero state. Thus, after the final word in the block, the PROM should read, "001XXXXX" next, in order to cause a reset and start the cycle again. A table for a typical PROM format is shown in Fig. 14. The PROM presently on the board is set up for 16 analog words in sequence followed by 16 digital words in sequence. An EOVI is tagged with the last digital word and a reset is issued at the end. This PROM is listed in hexadecimal in Appendix 4. Note that if several different

<u>DATA</u>	<u>DENOTES</u>
0000XXXX	Analog Channel
0001XXXX	Digital Channel
001XXXXX	Reset Flag (after last word)
110XXXXX	EOS Interrupt
111XXXXX	EOV Interrupt (end of block)

TYPICAL PROM CODE

<u>ADDRESS</u>	<u>DATA</u>	<u>WORD</u>
00000	0000XXXX	First Analog Channel
00001	0000XXXX	Second " "
00010	0000XXXX	Third " "
00011	0000XXXX	Fourth " "
.	.	.
.	.	.
.	.	.
10000	0001XXXX	First Digital Channel
10001	0001XXXX	Second " "
10010	1111XXXX	Last Channel (EOV)
10011	001XXXXX	Reset; End of data
10100	XXXXXXXX	Unprogrammed

Figure 14. PROM Format

patterns want to be tried for test purposes, they can all be stored at different addresses and the address lines A9 and A8 to the PROM, which are presently grounded, may be strapped for the proper pattern.

4.3 Physical Layout

The vehicle multiplexer circuit in schematic TT-8 is built on a Motorola EXORCISOR-type wire wrap card, like the transmitter. The multiplexer is wire wrapped with the same ground and five-volt supply matrix for low noise as before. The interface to the shared RAM, as well as the plus five and plus and minus 12-volt supplies enter the board on the card edge connector. A minus five volt, three-terminal regulator (7905) runs on the minus 12-volt supply for use by the EPROM.

Two 50-pin ribbon cable connectors provide input/output signals to the card. The input connector, shown on the left hand side of the board, carries 14 analog channel inputs, as well as the 16 digital select lines from the decoder, U5. Additionally, the plus and minus 15 volt supplies for the analog circuitry are brought in from the vehicle supply via this connector. The three-state data bus from digital channels also comes in at this point, where it is fed to the input of the FIFOs.

The right hand, or output connector, is used to tie the data output from the FIFOs to the three-state telemetry transmitter bus. This bus includes 16 address lines and 16 data lines (32 total), as well as the FIFO control lines shift out, output ready, and output enable. This arrangement is pin for pin compatible to the laser FIFO connector, as well as the microprocessor PIA connector for diagnostic purposes (see

section covering diagnostics). A diagram of the physical connections on the rover is shown in Appendix A5.

The gain setting potentiometers for the 16 analog channels are located on the left hand side of the board. The gain and zero (offset null) pots for the analog to digital converter are located on the top center of the board. The board layout is shown in schematic TT-10 and a photograph of the board is in Appendix 7.2.

PART 5

TELEMETRY RECEIVER

The telemetry receiver is shown in block diagram form in Fig. 15. When used on the dynamic test platform, serial data and clock from the transmitter are sent directly to the line receivers via a baseband coaxial link. When the system is implemented on the autonomous roving vehicle, however, the data is first received and demodulated from an RF carrier. Thus, an RF receiver is employed, followed by a demodulator/clock recovery network.

The ADLC at this end is run in the receive mode. Initialization of the control registers is done at start up as in the transmitter. The data from the ADLC is demultiplexed back into its original 16 bit address by 16 bit data format and shifted into a FIFO stack on the general-purpose interface board (GPIB). The GPIB resides in the Prime 750 computer mainframe and its operation is described in Donaldson's paper [5].

5.1 ADLC Configuration

The data link controller chip in the receiver is set up in much the same manner as the chip in the transmitter of Section 3.2. The main difference is that here control register 1 is set so that the transmitter portion stays reset (bit 7 set). Again, the receiver is put in the prioritized status mode and receiver data service request mode (RDSR). This causes interrupt requests to be inhibited and the RDSR line will go high when data from the receiver is ready to be read.

The priority of the status is such that any conditions implying

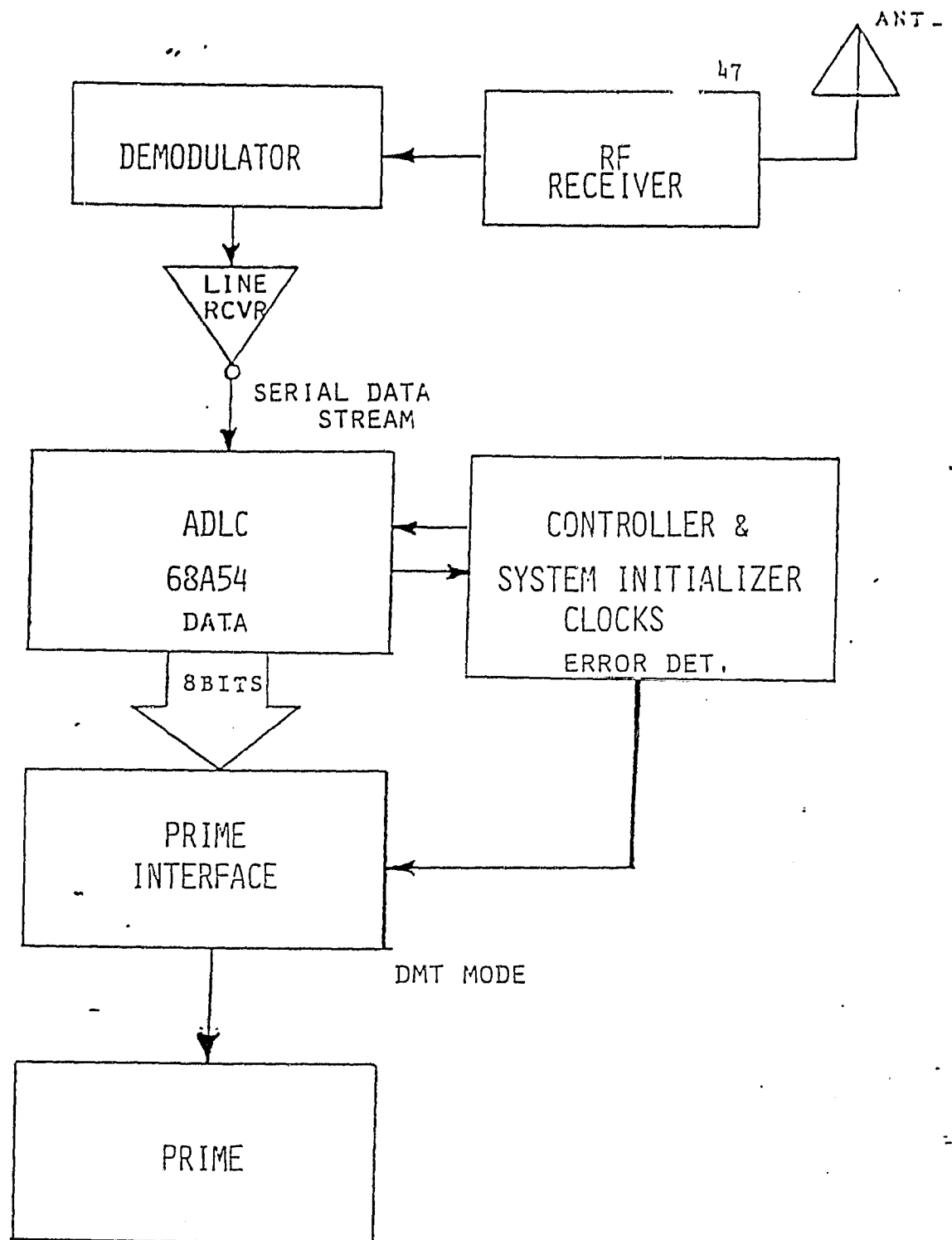


Figure 15. Telemetry Receiver

a bad or mis-received word (e.g., error, receiver overrun or abort conditions) are given the most importance. The RDSR or receiver data available (RDA) condition is given a lower priority. Thus if the status register is constantly polled, the error conditions will mask out the other status so that the word may be neglected and we can wait for the start of the next frame.

5.1.1 Initialization Coding. A table of the receiver initialization PROM coding is shown in Appendix 6. Addressing starts at location zero. PROM 1 holds the data and PROM 2 specifies the address of the control register. Notice also that this PROM controls the ADLC's chip select line, as well as the START signal, which takes control out of the initialization mode.

5.2 Hardware Description

The initialization hardware used is similar to that used in the telemetry transmitter. Upon system reset, master reset (\overline{MR}) causes all counters and flip-flops to initialize.

The initialization hardware is shown in schematic R-1. Flip-flops 16K wake up in the INIT mode after "master reset" is issued. INIT causes the ADLC data and address bus to be tied to the 8223 open-collector, initialization PROMs. Binary counters running synchronously from the 1 MHz ϕ_1 are used to address the PROMs. Clock ϕ_2 , which is the logical invert of ϕ_1 , is used to drive the ADLC "E" clock. Since data is set up on the PROMs at the rising edge of phase ϕ_1 , and subsequently clocked into the ADLC on the rising edge of phase ϕ_2 , the data at the FROM addresses initializes the device.

At the end of the initialization routine, data line E_5 of PROM 20k causes 16 k-A to be set when ϕ_2 rises. This latches flop 16 k-B to the $\overline{\text{INIT}}$ or non-initializing mode. It is at this point where we want to enter the receiver control loop. This was accomplished when flop 16 k-B just toggled, causing one shot 47k (schematic R-2) to be triggered. A one shot is used to enter a bit into the ring-counter type controller. The receiver control section, a multiple-state sequential controller, is also shown in schematic R-2.

A word must be mentioned about the clocks used in the receiver. Since the controller makes decisions based on the bits of the hardware status register, and this status register is latched by the rising edge of ϕ_1 , a delayed version of ϕ_1 must be used to clock the sequencer. If this were not done, then a race condition would exist at the time when the status register and the sequencer were simultaneously clocked. Also, since the sequencer sets up addresses for the ADLC to act on, and ϕ_2 clocks the ADLC, this phase cannot be used to clock the sequencer, either. Instead, a delayed phase, ϕ_1 , called $D\phi_1$, is generated with some gate propagation delays. Thus $D\phi_1$ occurs about 60 nanoseconds after ϕ_1 , being delayed by the propagation time through six 7404 inverters. This provides enough setup time for the status register conditions to appear at the data inputs to the sequencer flops, as they only require a 20 nanosecond setup time (60 nanoseconds is still enough time even including the single gate delay involved in decoding the status bits). Note the delay time is not critical, as long as it lies between ϕ_1 and ϕ_2 .

The other clocks important to the receiver are the data read clocks, DATAREAD A and DATAREAD B. DATAREAD A occurs for one period of ϕ_1 only when a "read status register 2" operation occurs. This clock is used to latch the ADLC status into the hardware status register 41L, schematic R-3. This clock is generated by 14-M, a flip-flop.

DATAREAD B occurs for one ϕ_1 period when a "read receiver FIFO" operation occurs. When the receiver FIFO is read from the ADLC, the current received byte is put on its data bus. It is DATAREAD B which clocks this byte into one of the four demultiplexer latches (41k, 33k, 31k or 29k). The above clocks are summarized in Fig. 16.

5.2.1 Controller. The control logic is shown in schematic R-2. The level set up by the one shot 47k after initialization gets clocked to setup state one. Actions are taken by incrementing to the next state flip-flop, contingent upon the states of the status that are gated. For example, at state 2 we test status register bit zero, which tells whether the first byte is present from the receiver. If SR bit 0 is true, then gate 35-k06 will be high, causing us to advance into state 3 on the next $D\phi_1$ rising edge. We can then proceed to read the receiver FIFO. If the SR bit 0 was not true, then gate 35-k08 would have gone high, looping us back into state one on the next $D\phi_1$ rising edge.

The flow chart for the control loop is shown in Fig. 17. The address present condition tested in state 2 means that the first byte of a frame is available to be read. Thus we sit in a tight hardware loop on this condition, guaranteed to acknowledge this condition with a minimum of delay. When we finally see this bit set, we go to state

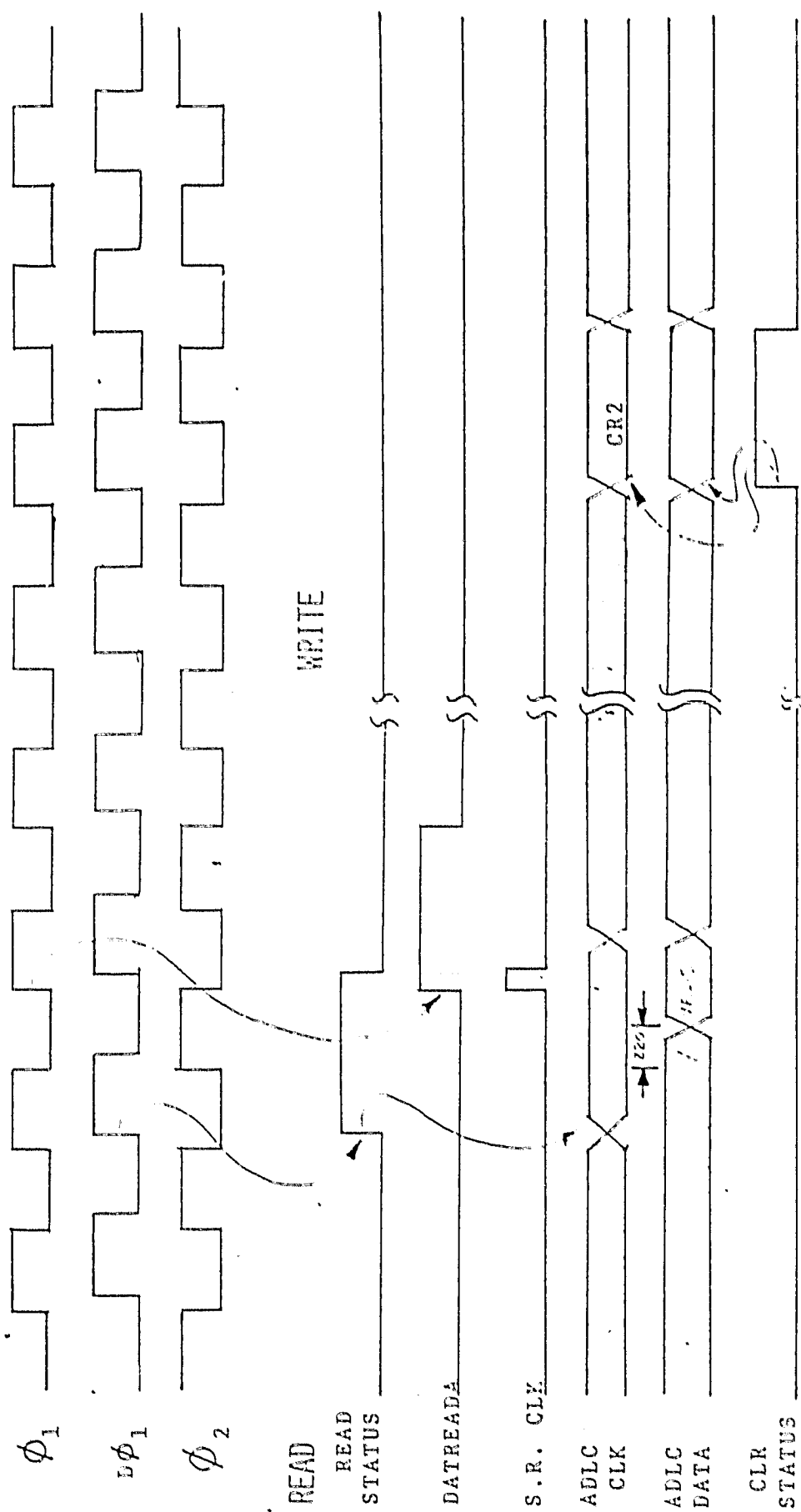


Figure 16. Receiver Clocks

three where we read the receiver FIFO and latch this byte with DATAREAD B into the first latch (41k). This is because the BYTE CTR (27L) selects the first latch to be clocked. Next, at state four, the byte counter is incremented to prepare for the triggering of latch 2 (33k).

The status register is once again read and tested. If the receiver data available bit is set, we service the ADLC FIFO to get the next byte. If, however, an error condition had occurred, we reset the byte counter, clock the error counter (which keeps track of the number of errors for performance information), and reset the status register.

It is important to reset the status register after testing it so that it will reflect current status the next time it is read. After an error, we return to the start of the loop without transferring any data to the interface. If we were waiting for the last byte to be ready, then once the frame valid status bit were set in state 6, we would proceed to state 7. There the receiver FIFO is read and the byte gets loaded into the last latch (29k). Next, in state 10, the first pair of FIFOs on the GPIB get shifted in with the 16 bit data word. Following that, in state 8, the other two GPIB FIFOs get clocked and the 16 bits of address get shifted into them. Due to the FIFO control circuit on the GPIB, all four FIFOs cannot be shifted in simultaneously or there will be an instant when all the FIFO output ready's will go high, causing a FIFO INTERRUPT condition to exist with GPIB status register (see Donaldson's paper [5]).

After the FIFOs are loaded, we prepare to receive the next frame by clearing the byte counter, resetting the ADLC status register,

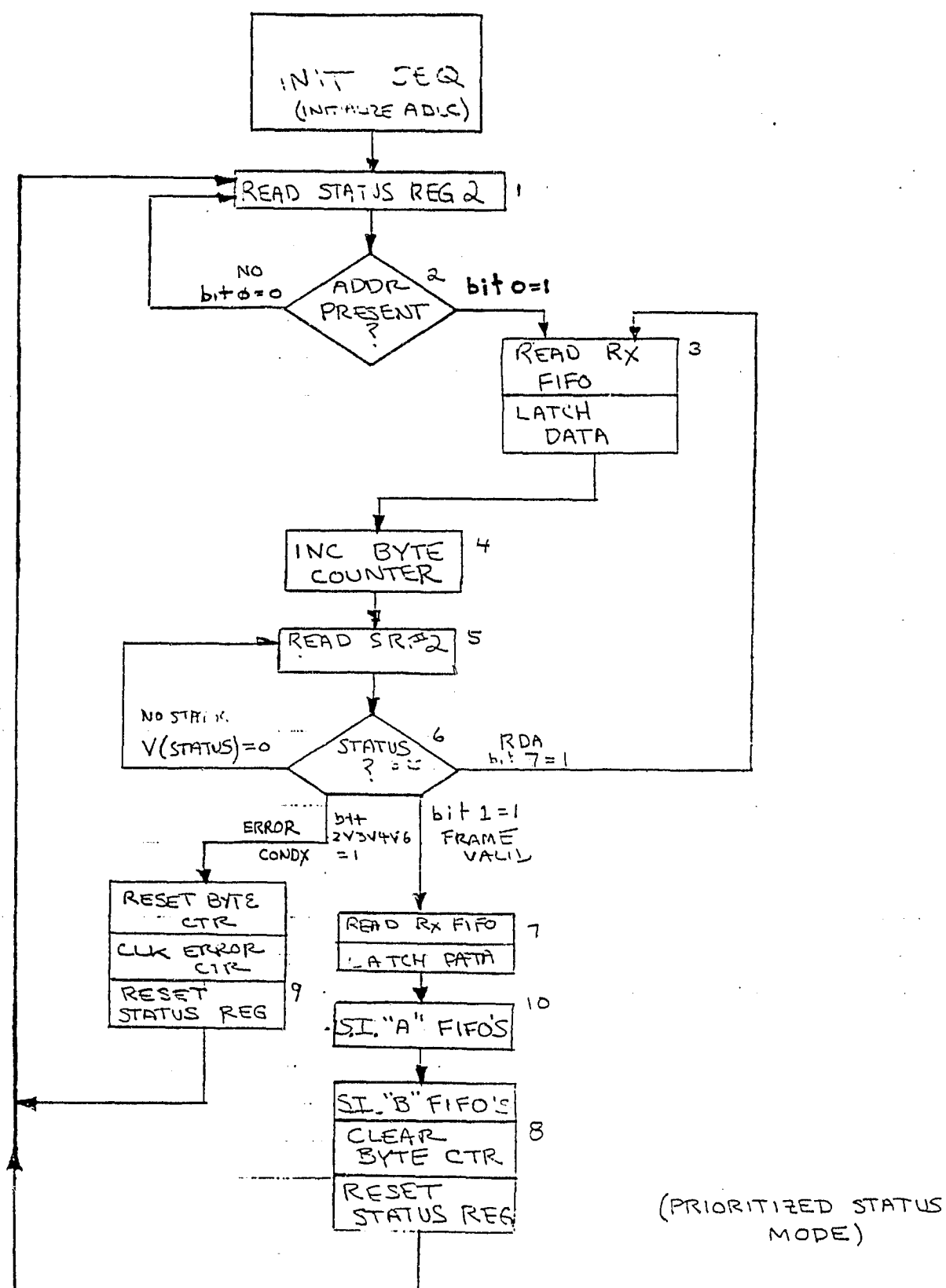


Figure 17. Hardware Control Loop

and re-entering state 1. If, when in state 6 none of the status bits were set, then we would simply sit in a tight loop between states 5 and 6, reading and testing ADLC status register 2.

Clearing the status register in the ADLC is accomplished by writing to control register 2, bit 5. This is done by the hardware in the following manner. The "reset status register" signal sets up the address of control register 2 on the ADLC through gates 12M, 31L and 27k on schematic R-3. Also, this signal enables data bus drivers 12k, which write the word "01100001" to control register 2, thereby clearing the status bit and leaving CR2 as it was. Note also that during a read receiver FIFO operation, the addressing encoder (gates) on schematic R-3 set up the proper ADLC address.

5.3 GPIB Signals

Since the receiver logic resides on the Prime general purpose interface board, many of the signals there are used. Five volt supply is taken from the card, distributed by a ground and power plane on opposite faces of the circuit board. R-4 shows the layout of the receiver on this board.

One master clock is used by the receiver. This clock is derived from the computer-generated 5 MHz BPCFCLK, available on the GPIB. A unique divide by five circuit (see Kennedy's paper [6]) is employed to yield the 1 MHz ϕ_1 clock. In this manner, all receiver operations are deterministic and synchronous.

The master reset signal (MR) used by the receiver is derived from the logic OR of two sources (16L). One source is the GPIB signal

SYSCL⁻, which is a reset signal generated by the computer hardware (e.g., power up, restart, etc.). The other, MASCL⁺, is generated in software whenever the GPIB board is assigned. Thus it is good practice to assign the rover GPIB board before each use in order to start the receiver loop.

The data and shift in signals are sent to selectors in the GPIB control circuitry (see Donaldson's [5] paper). These selectors decide whether the data that the GPIB FIFOs get comes from telemetry or the GPIB data emulator. The emulator is used for generating data in the diagnostic mode. In normal operation the receiver control circuitry will toggle the shift-in inputs via the selectors.

5.4 RF Receiver

The PAM (pulsed-amplitude modulated, or in this case, equivalently, the amplitude shift keyed) carrier must be received, converted to baseband, and demodulated before it can be used by the above digital circuitry. Good sensitivity and a wide enough bandwidth are of prime concern in these stages. A diagram of the RF section is shown in schematic R-5.

5.4.1 Front End. The signal from the antenna is fed into the RF amplifier and mixer located in the tuner. The resulting signal is fed to the intermediate frequency (IF) amplifier. The tuner and IF amplifier are commercially built units from a Quasar television receiver. These were selected because they were readily available, inexpensive, tunable and allowed the necessary bandwidth. The noise

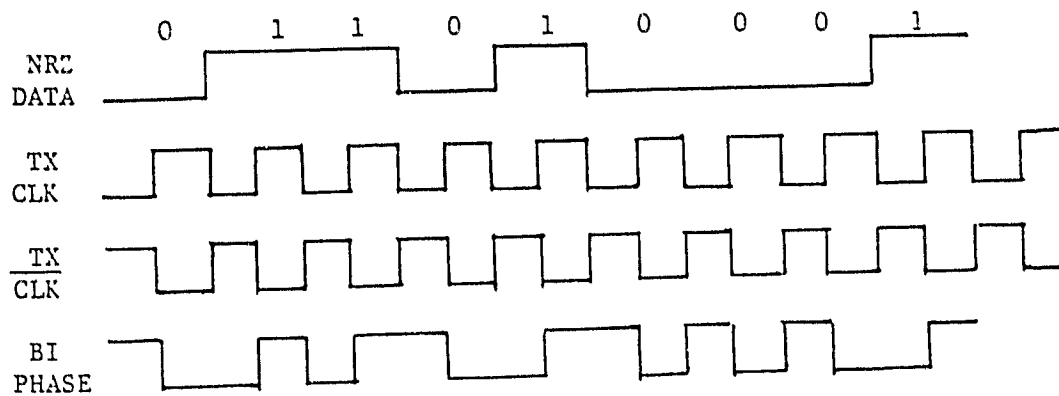
figure for the RF portion is approximately 12 dB, which is not at all exceptional, but is small enough for our purposes, especially with the signal levels used. The last stage of the IF amplifier contains the detector. Envelope detection is performed on the PAM carrier.

5.4.2 Demodulator. The detected baseband signal from the front end is now converted to digital levels with comparator U1. Hysteresis is employed here to minimize errors due to noise.

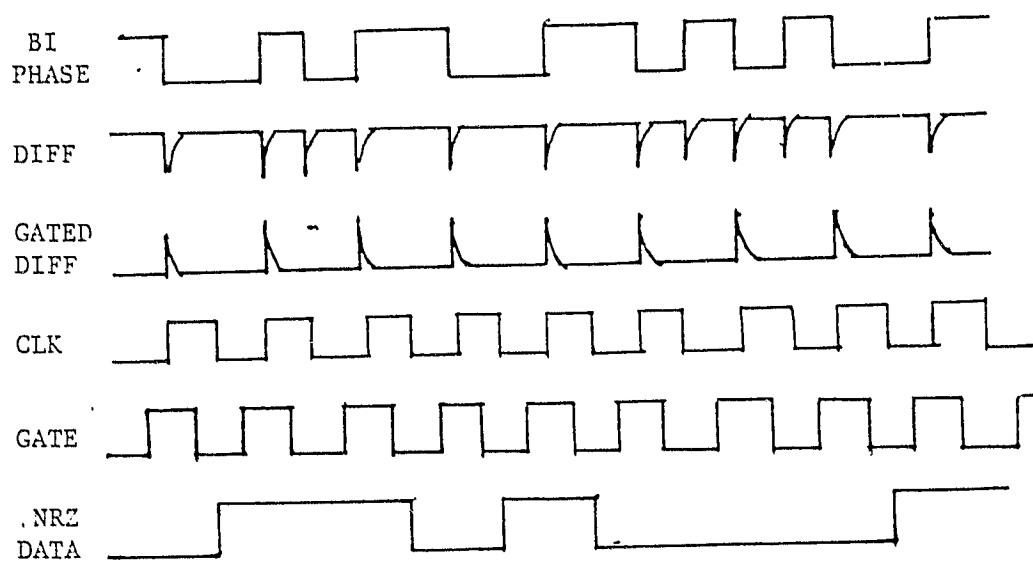
Once in a TTL-compatible form the serial biphase data stream must be fed to a carrier recovery loop in order to retrieve the data clock. Also, the biphase data must be converted back into the NRZ-format data stream.

The biphase data stream is fed into a differentiator made with an exclusive-or gate and a capacitor. This marks the locations of data transitions, providing a reference at twice the bit rate. The timing diagram (Fig. 18) shows where there will be missing pulses corresponding to where the data changed states in the serial stream. The VCO (voltage controlled oscillator) in the phase-locked loop (PLL) runs with the differentiated signal above at the reference. The VCO is forced to run at twice the data bit rate due to a divide by two flip-flop (U2A) in the loop to the phase comparator. The input to the phase comparator is gated so that only every other differentiated pulse gets to it. Because of this the phase-locked loop does not see the missing pulses, and the VCO runs at twice the clock frequency.

Flip-flop U2B uses an inverted clock and output from U2A to provide a clock called "GATE" which is shifted in phase by 90° . This signal is used to clock a flop (U3A) to recover the NRZ bit stream.



a. Transmitter



b. Receiver

Figure 18. Demodulator Timing

When the transmitter is first powered up, it sends a string of alternating ones and zeros during the master reset time (25 msec). This is done so that this phase-locked loop gets initially synchronized. This guarantees differentiated pulses at each clock transition, in order for the loop to start up in lock.

After recovering clock and NRZ data, these signals are fed to line drivers where they drive a coaxial line to the receiver located on the GPIB board in the computer mainframe.

5.4.3 Antennas. The antenna used for receiving is a broad band Yagi TV-type antenna. This antenna is presently mounted on the roof of the Jonsson Engineering Center and feeds a 50 ohm coaxial cable via a matching transformer. This antenna has a gain at the operating frequency of about 9 dB. The coaxial cables from the roof are diagrammed in Fig. 19. The antenna is being used for both the command receiver and the telemetry receiver. Because of the 3 dB lost in a splitter and the RG-8U coaxial cable to the roof having a loss of 1.5 dB, this leaves an overall gain before the receiver of 4.5 dB.

5.4.4 Location. The RF receiver is located in the rover lab, Room JEC 3210. The antenna cable from the roof (splitter), as well as the clock and data lines to the computer interface, connect to the receiver box as indicated. A 20 volt supply is connected to the receiver box via a phono plug.

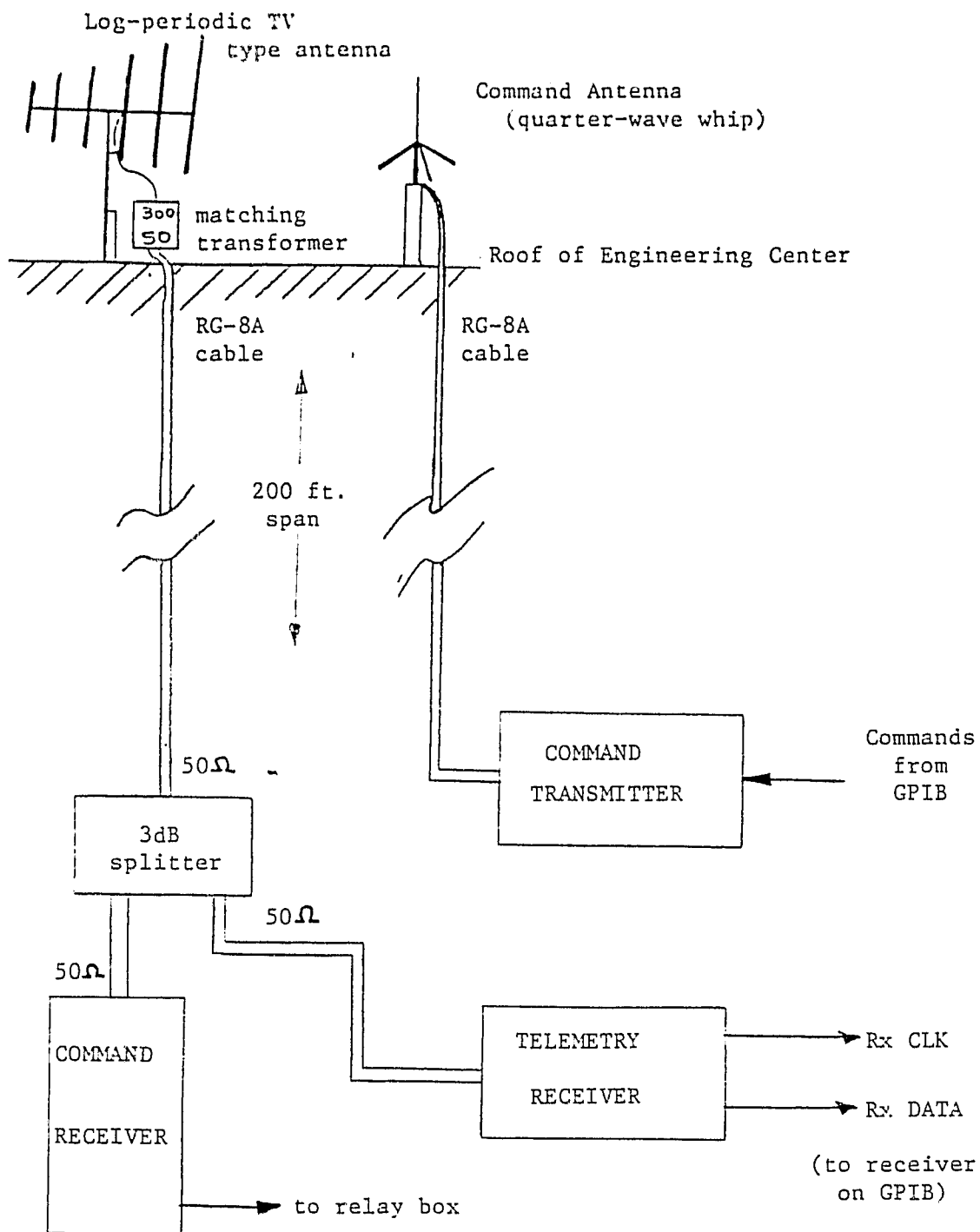


Figure 19. Antenna and Cable Arrangement

PART 6
PERFORMANCE

6.1 Calculations

The performance of the telemetry link has been both calculated and tested.

The most important calculation from the standpoint of the largest probability of error is the feasibility of the RF link. Path loss, power levels, antennas and bandwidth all affect the signal to noise ratio at the receiver. Also, the type of modulation used indicates how much margin is needed for a low error rate. Carlson [7] gives some equations for signal to noise ratios and error probabilities. I have made some first order calculations to determine the feasibility of the link described and found plenty of margin.

Assuming a maximum range of one mile, which is more than necessary for testing the vehicle within the campus, the signal to noise ratio is calculated using the specifications given. Assuming only line of sight radio propagation, and not counting secondary reflections or ground effects, we get a power-out to power-in ratio described by [7].

$$\frac{P_{out}}{P_{in}} = G_{TA} G_{RA} \left(\frac{\lambda}{4\pi l} \right)^2$$

where G_{TA} = Gain in the transmitting antenna

G_{RA} = Gain in the receiving antenna

λ = Wavelength of the carrier operating frequency

l = Distance

Expressed in decibels, the loss is

$$L_{dB} = 22 + 10 \text{ LOG } \left(\frac{l}{\lambda}\right)^2 - (G_{TA} + G_{RA}).$$

The receiving antenna gain overall is 9, minus 3 dB for the splitter and 1.5 dB for the coaxial cable, or 4.5 dB. The transmitting antenna has a gain of 3 dB. For a $\lambda = c/f_0 = 5.56$ meters, and a range of one mile (1600 meters), we get

$$\begin{aligned} L_{dB} &= 22 + 10 \text{ LOG } \left(\frac{1600}{5.56}\right)^2 - (3.0 + 4.5) \\ &= 64 \text{ dB.} \end{aligned}$$

Next the effective noise power at the input to the receiver must be determined. If the receiver noise figure is 12 dB and

$$\text{N.F.} = 1 + \frac{T_{EQ}}{T_0},$$

then the equivalent noise temperature due to the receiver is 14.85 times that of T_0 (operating temperature, 273 °K), or 4054 °K.

The equivalent noise power, N , is then found from

$$N = kT_{eq}B$$

where k = Boltzman's constant = 1.37×10^{-23} j/°K

For a bandwidth of 1 MHz,

$$\begin{aligned} N &= (1.37 \times 10^{-23} \text{ j/°K}) (4054 \text{ °K}) 10^6/\text{sec} \\ &= 5.554 \times 10^{-14} \text{ watts.} \end{aligned}$$

The signal to noise ratio is [7]

$$\left(\frac{S}{N}\right)_D = \frac{S_T}{L_N}.$$

The transmitted power is one watt and L and N are given above.

$$\begin{aligned} \left(\frac{S}{N}\right)_D &= \frac{1W}{74 \text{ dB } (5.554 \times 10^{-14} W)} \\ &= 68.55 \text{ dB.} \end{aligned}$$

This is a fairly good noise margin at the receiver input and if the receiver, demodulator and treatment of the PAM signal do not introduce much error, then the RF link should perform extremely well.

Tests have shown that with the telemetry link on the test platform with a base band link (line drivers and coaxial cables between the transmitter and receiver), the system has performed very well. Detected errors have been on the order of 5 per hour. At 500 kb/s this is about 2.778×10^{-9} errors per second. This process will continue until word 64, containing an EOV interrupt in its most significant bit position, is sent. After this point, the process repeats, starting from address zero again.

Other diagnostic programs are available, and the reader should consult Doig's paper [8] for them.

6.2 Diagnostics

Diagnostic programs for the system have been written both on the Prime computer and on the steering and propulsion microprocessor..

The microprocessor has been used to debug the prototype receiver. Since the actual receiver is no longer controlled by the microprocessor, those programs will not be discussed here. One program which is important, however, is the routine which generates test

data for the transmitter to send.

This program repetitively sends a known pattern of data to check for errors through the system. The microprocessor code is listed in Appendix 9. Also, the program is listed on a cassette tape for quick entry into microprocessor memory.

To use this program, one first loads it into memory. The starting address is hexadecimal "0000." The microprocessor will send data out of its PIA (parallel interface adaptor). After receiving a shift out via a PIA input port, the microprocessor will put a 16-bit address, 16-bit data word out on its PIA. The first word is address zero, data = zero. The second word is address = 1, data = 1, and so on.

PART 7

CONCLUSIONS

Throughout the design process, flexibility for future expansion has been considered. Three state busses are employed to provide for future addition of other hardware components, or even a microprocessor controller. Also, 16 bit address and data words are used for a greater number of possible channels.

The number and order in which vehicle data can be sent may be readily changed for different configurations by plugging in different PROMs.

The telemetry system has been used in the lab on the scanning mast platform. This has provided a good test set-up as well as a means of passing platform data. It was here that actual performance measurements have been made.

A few conclusions have been reached regarding the use of a time-shared machine such as the Prime for the real-time control application of the Mars Rover. Throughput is significantly reduced with many users on the system. As few as six users will start to limit performance at the current data rates.

Although the information is being loaded into the computer in a direct memory access mode, a heavy load will cause an insufficient number of transactions to the GPIB buffers. When this happens, FIFO interrupts will occur, signifying that the GPIB buffers are getting overvilled.

The best cure for this problem is to reduce the telemetry data

rate, which may be a sacrifice to laser data. A safety margin has been included in the data link design in order to ease this constraint. Thus the data rate can be reduced by providing an external transmitter clock.

Another remedy is to reduce the number of interrupts processed in the software and on the GPIB. Since the CPU must take time to service each interrupt, a high interrupt rate will also slow the system down and reduce the time available to service the GPIB FIFOs. This was found to be the case on the back of scan when end of vehicle interrupts were sent after each block. Care must be taken to limit the number of interrupts that are serviced in the software if real-time control is desired. Enough information should be gained by simply regarding the end of scan interrupts.

In addition, much of the telemetry link capabilities are being squandered when the GPIB strips out the upper address bits. The laser data is being sent with a complete address, which can be used to map the information directly into memory, instead of using software counters that are keyed on end of azimuth interrupts. Running in this mode would require a little bit more memory space (5k block) but would yield a maximum throughput, leaving the CPU unbettered by the DMA transactions.

In any event, the system has capabilities for future expansion and variable data rates.

Integration of the system onto the roving vehicle has been eased through the use of an interconnecting box. All I/O lines are available at this box for diagnostic and modification purposes. Also, connections to the vehicle meet at this box.

The system built on the 6800 card cage should prove to be a versatile and effective data link.

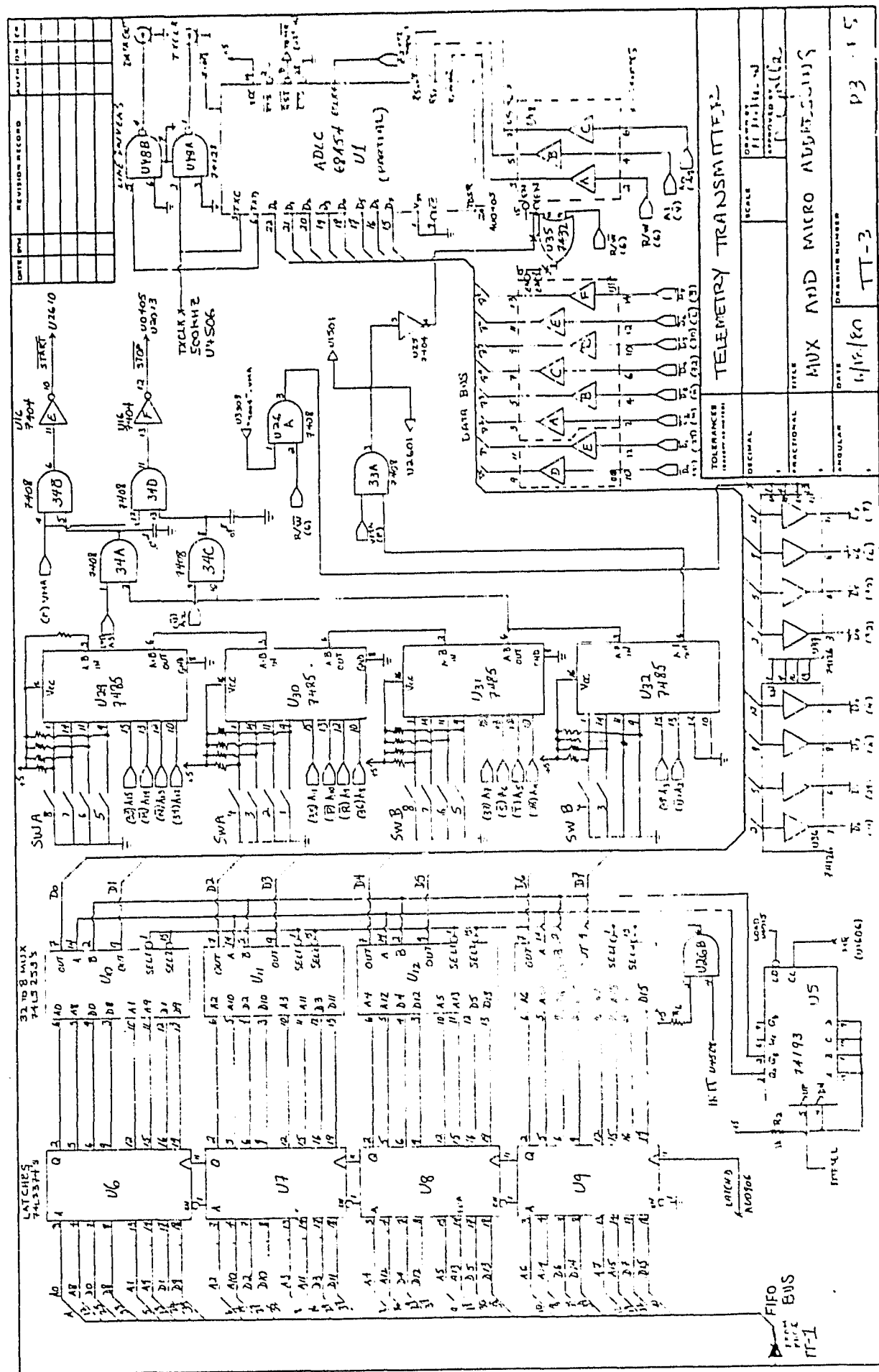
PART 8

REFERENCES

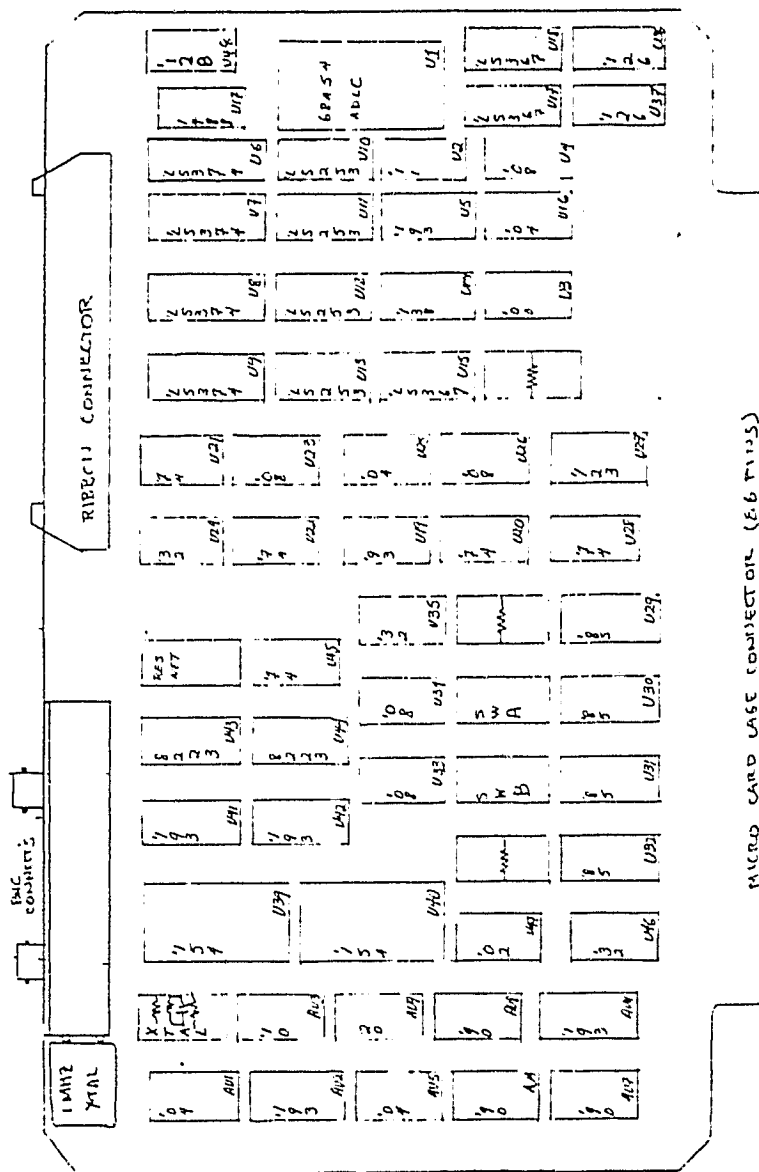
1. Craig, J., "Design and Implementation of the Laser Scanning/Multi-Detector Controller for Hazard Detection System," RPI Technical Report MF-59, Troy, N.Y., August 1978.
2. Motorola, Inc., "The Complete Microcomputer Data Library," 1978, pp. 1-233 to 1-255.
3. Bogdan, D., "A Propulsion System for the Mars Roving Vehicle," RPI Technical Report MF-70, Troy, N.Y., August 1980.
4. Fairchild Semiconductor, "MOS and CCD Data Book," F3351 First in, First out Memory, Mountain View, Calif.
5. Donaldson, J., "Laser Optical Appraisal and Design of a Prime/Rover Interface," RPI Technical Report MF-68, Troy, N.Y., December 1979.
6. Kennedy, William, "Control Electronics for a Multi-Laser/Multi-Detector," RPI Technical Report MF-73, Troy, N.Y., August 1980.
7. Carlson, A.E., Communication Systems, McGraw-Hill, Inc., 1975, pp. 137-138, pp. 389-400.
8. Doig, G., "Electronics and Software Subsystems for an Autonomous Roving Vehicle," RPI Technical Report MF-76, Troy, N.Y., August 1980.

PART 9
SCHEMATIC DIAGRAMS





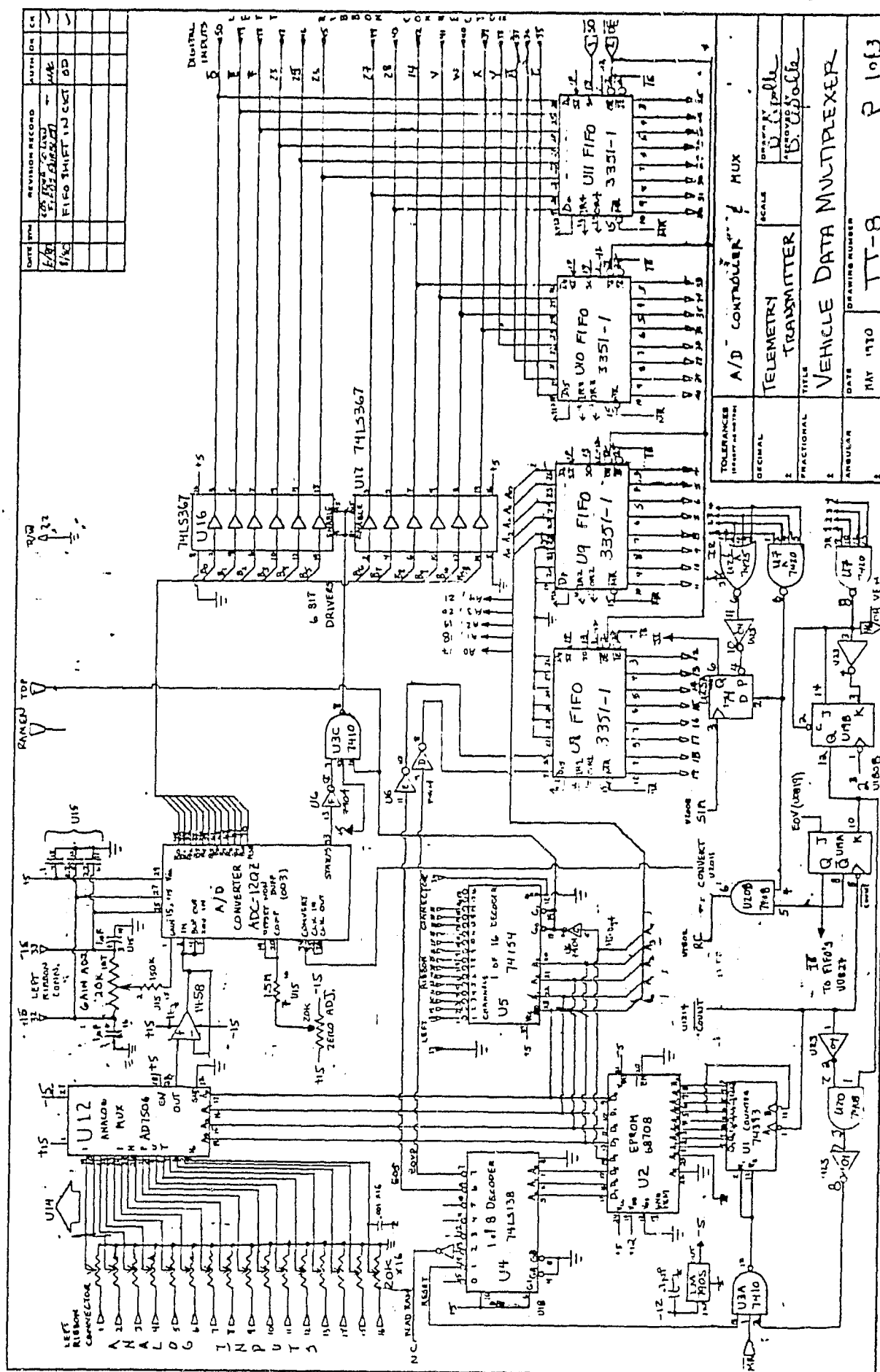
ORIGINAL PAGE IS
OF POOR QUALITY



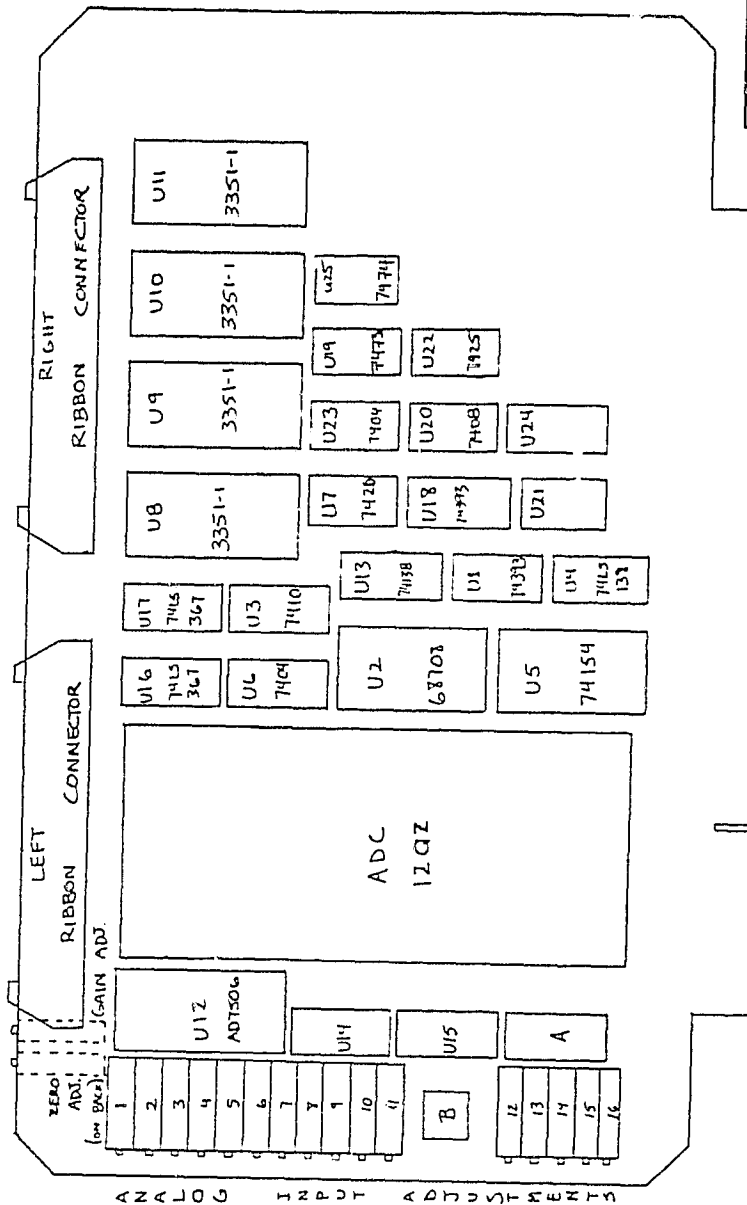
DATE	REVISION	RECORD	DATE	REVISION	RECORD

TOLERANCES UNLESS OTHERWISE SPECIFIED	DECIMAL	FRACTIONAL	ANGULAR	DATE	DRAWING NUMBER	TITLE	SCALE	QUANTITY OF REVISIONS	REVISIONS
	1	1	1			TELEMETRY LINK CONTROLLER CHIP LAYOUT (COMPONENT SIDE)		1	1

TT-4 p 4 of 5

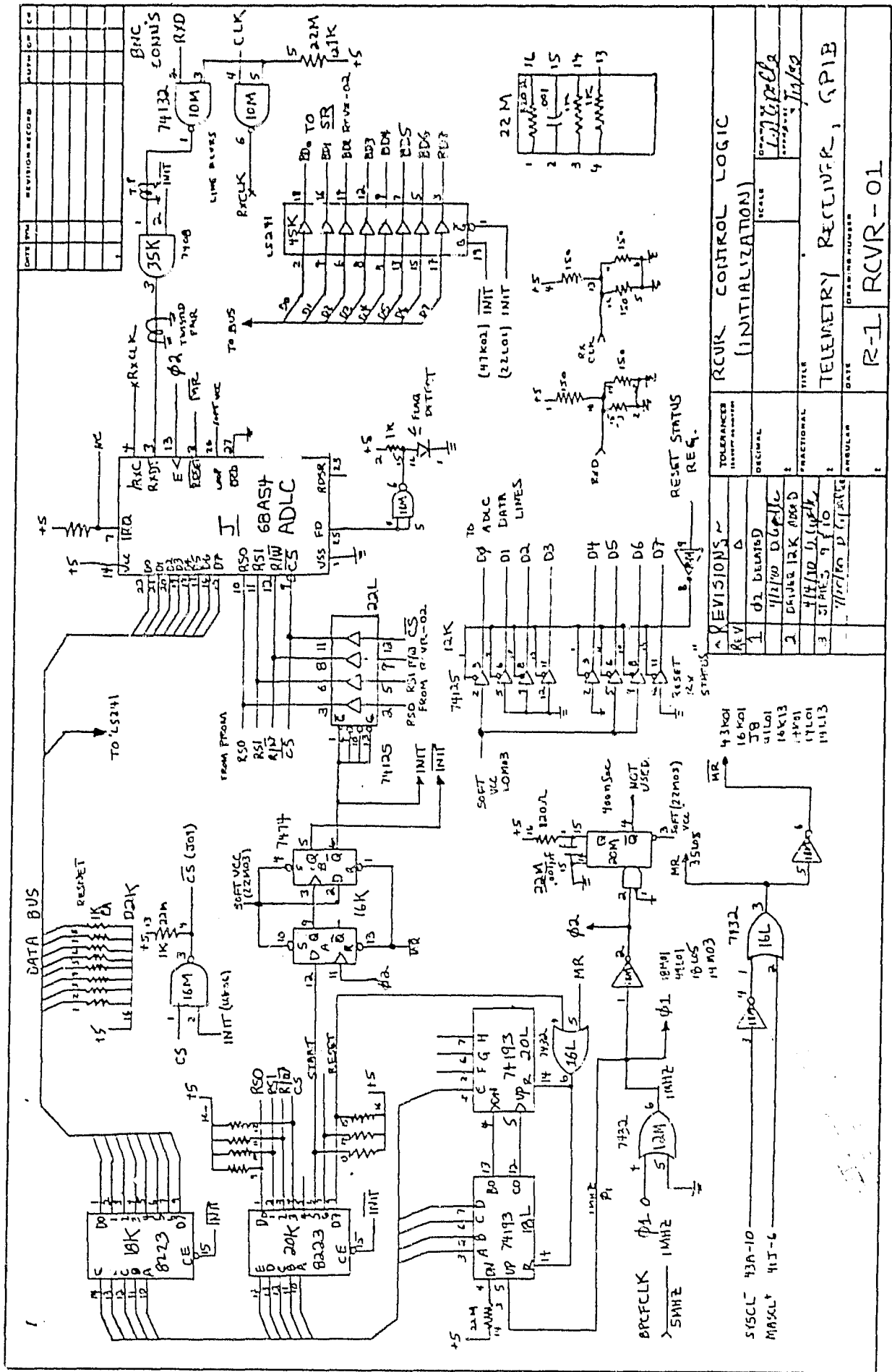


COMPONENT VIEW



TOLERANCES UNLESS OTHERWISE SPECIFIED			
DECIMAL	FRACTIONAL	ANGULAR	PLACEMENT
1	1	1	1
HUX COMPONENT LAYOUT			
TELEM XMRK		SCALE	GRAPHIC SCALE
TITLE		DATE	REVISION
VEII DATA MUX		MAY 20	TT-10
DRAWING NUMBER		1	

DATE	BY	REVISION	RECORD	DATE	BY
5/20/80	MC	Added 12 pin			



PART 1C
APPENDICES

- A1 List of Interrupts
- A2 Transmitter Initialization PROM Coding
- A3 Microprocessor-Controlled Transmitter Flow Chart and Code
- A4 PROM Table of 2708 presently on MUX Board
- A5 Physical Wiring between Boards
- A6 Receiver Initialization PROM Coding
- A7 Photos:
 - A7.1 Telemetry Transmitter Card
 - A7.2 Vehicle MUX Card
 - A7.3 Interconnecting Box and Card Cage
 - A7.4 Card Cage Mounted on Rover
 - A7.5 Autonomous Roving Vehicle
- A8 Microprocessor Diagnostic Program
- A9 Glossary of Terms

<u>BIT</u>	<u>ABRV.</u>	<u>NAME</u>	<u>SOURCE</u>	<u>PURPOSE</u>
A13	EOA	End of Azimuth	Laser	Indicates the last azimuth shot.
A14	EOS	End of Scan	Laser	Indicates the last shot of a scan.
A15	EOV	End of Vehicle	Veh Mux	Indicates the last vehicle word; signals the end of a vehicle block.
a. Transmitted				
A12	FOFL	FIFO Overflow	GPB	Occurs when the FIFO is not getting serviced by the computer fast enough.
A11	TO	Time Out	GPB	Occurs if there has not been any interrupts (received data) during the last 267 msec.

b. Received

Appendix 1. Table of Interrupts

CR1 (RS1, RS0, R/W = 0,0,0)

b ₀	0	Address Control
b ₁	0	Rcvr IRQ enable
b ₂	0	Tx IRQ enable
b ₃	1	RDSR mode
b ₄	1	TDSR mode
b ₅	0	Rx frame discontinue
b ₆	1	Rx reset
b ₇	0	Tx reset

CR2 (RS1, RS0, R/W = 0,1,0)

b ₀	1	Prioritized status mode
b ₁	0	2/1 byte transfer
b ₂	0	Flag / mark idle (0 / 1)
b ₃	0	TDRA / FC (0 / 1)
b ₄	0	Tx last data
b ₅	0	Rx clear status
b ₆	0	Tx clear status
b ₇	0	RTS control

CR3 (RS1, RS0, R/W = 0,1,0, AC set)

b ₀	0	Logical control field
b ₁	0	Extended control field
b ₂	0	Auto address extend
b ₃	0	01 / 11 idle
b ₄	0	Flag detect status
b ₅	0	Non-loop
b ₆	0	Go active on poll / Test
b ₇	0	Loop on line / DTR

CR4 (RS1, RS0, R/W = 1,1,0, AC set)

b ₀	1	Opening / closing flags
b ₁	1	Tx word length
b ₂	1	select
b ₃	1	Rx word length
b ₄	1	select
b ₅	0	Tx abort
b ₆	0	Abort extend
b ₇	0	NRZ / NRZI (0 / 1)

PROM 1

PROM 2

ADDRESS				DATA								DATA								PE:		
E	D	C	B	A	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀		
0	1	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	CR1, reset
0	1	0	0	1	0	1	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	CR1, AC
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	CR3
0	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0	1	0	1	1	CR4
0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	CR1
0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	CP2
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	START
0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	PESET
(PROM)				(to ADLC data port)								(control)								(ADLC address)		

(PROM)

(to ADLC data port)

(control)

(ADLC address)

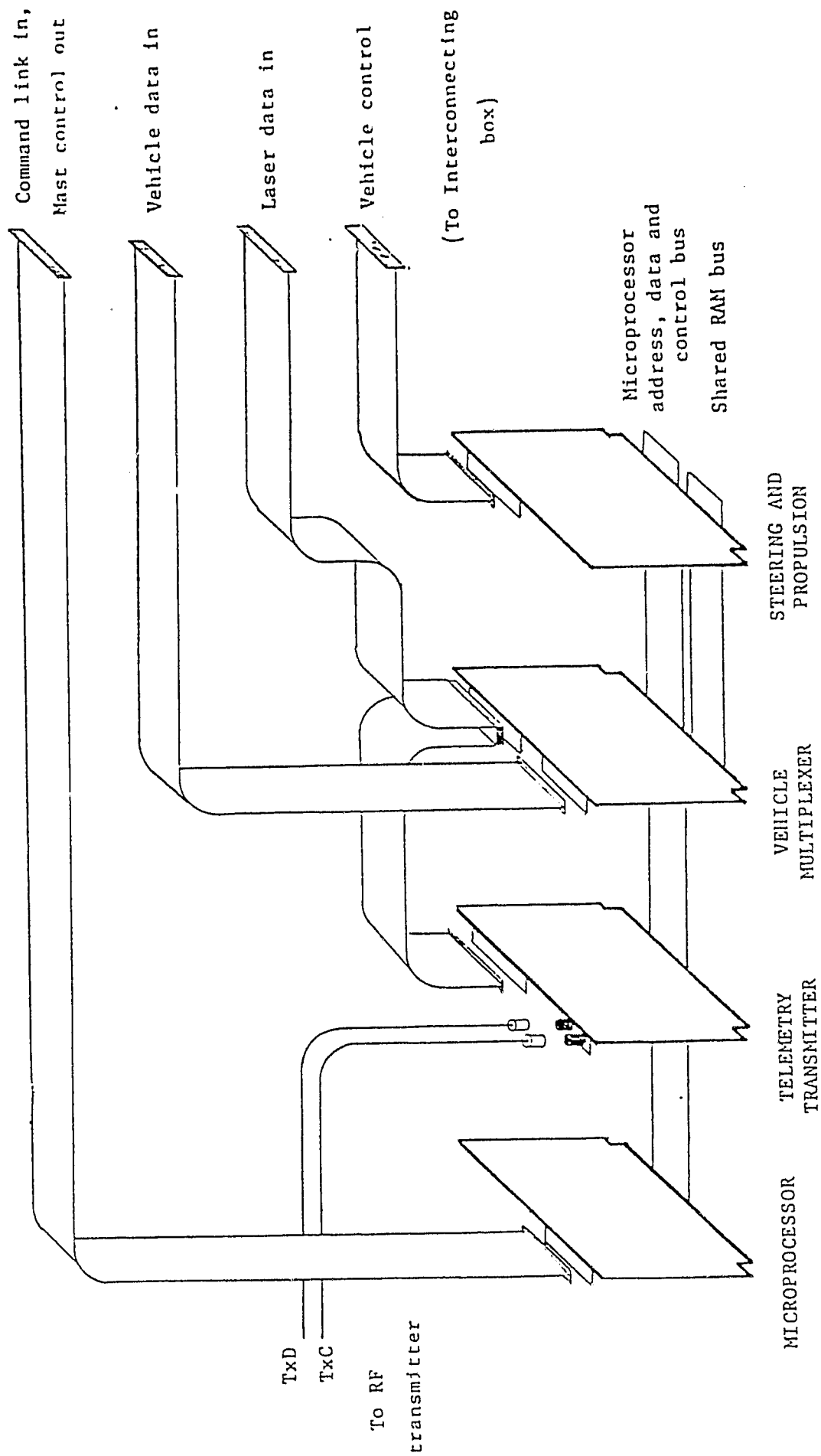
PROM used is an 8223.

Appendix 2.2. PROM table for transmitter

<u>ADDRESS</u>	<u>DATA</u>	
00	00	
01	01	
02	02	
03	03	
04	04	
05	05	
06	06	
07	07	
08	08	
09	09	16
0A	0A	Analog
0B	0B	Channels
0C	0C	
0D	0D	
0E	0E	
0F	0F	
10	10	
11	11	
12	12	
13	13	
14	14	
15	15	
16	16	
17	17	
18	18	
19	19	16
1A	1A	Digital
1B	1B	Channels
1C	1C	
1D	1D	
1E	1E	
1F	FF	EOV
20	20	Reset

PROM used is an M68708 (2708)

Appendix 4. Vehicle Multiplexer PROM Table

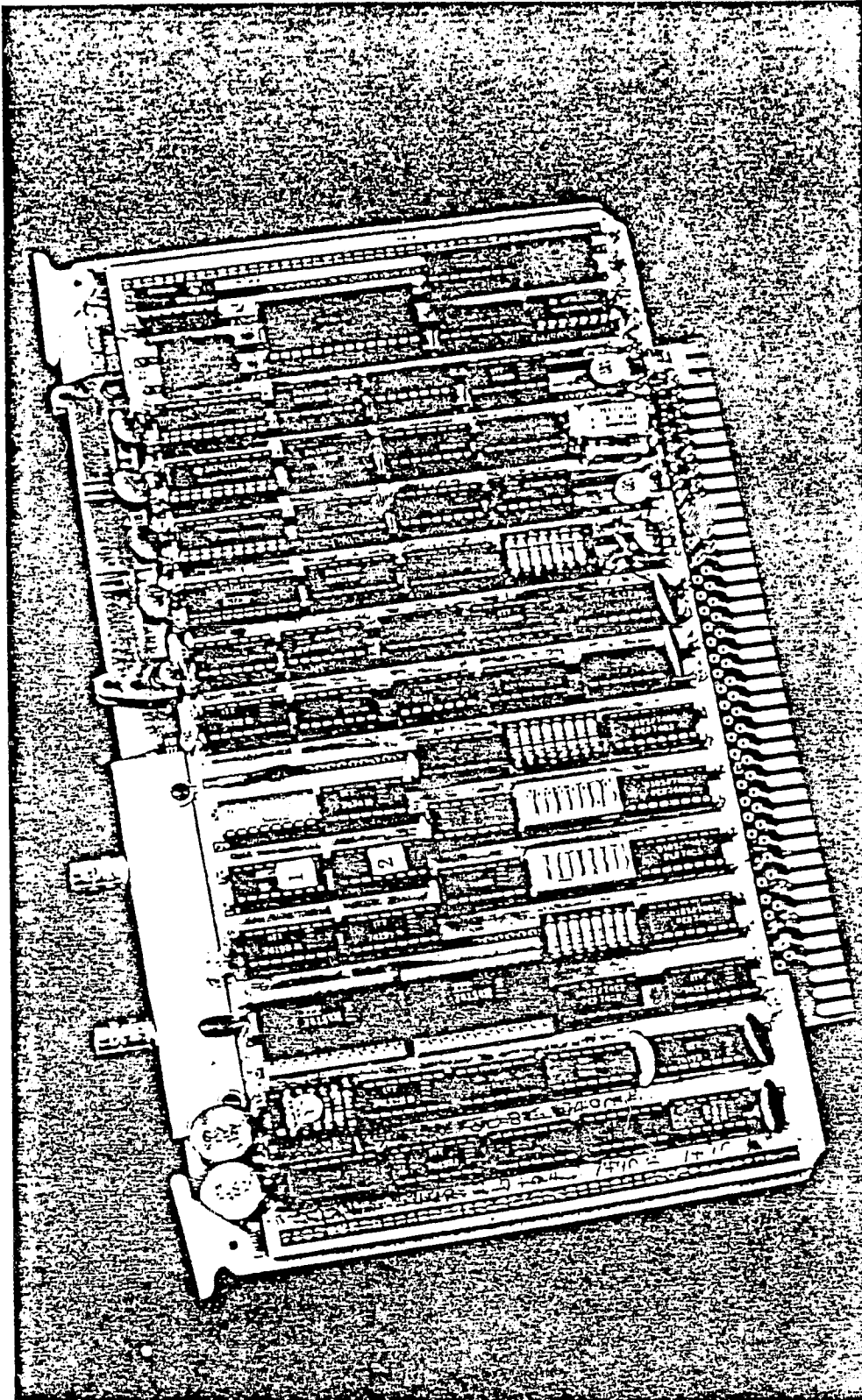


Appendix 5. Board Layout and Interconnection

PROM 1										PROM 2										RE:		
ADDRESS		DATA								DATA												
E	D	C	B	A	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀		
0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	CR1/reset/AC
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	CR3
0	0	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	1	0	1	1	1	CR4
0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	CR1
0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	CR2
0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	CR1
0	0	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0	0	0	START
0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	RESET
(PROM)				(to ADLC data port)										(control) (ADLC address)								

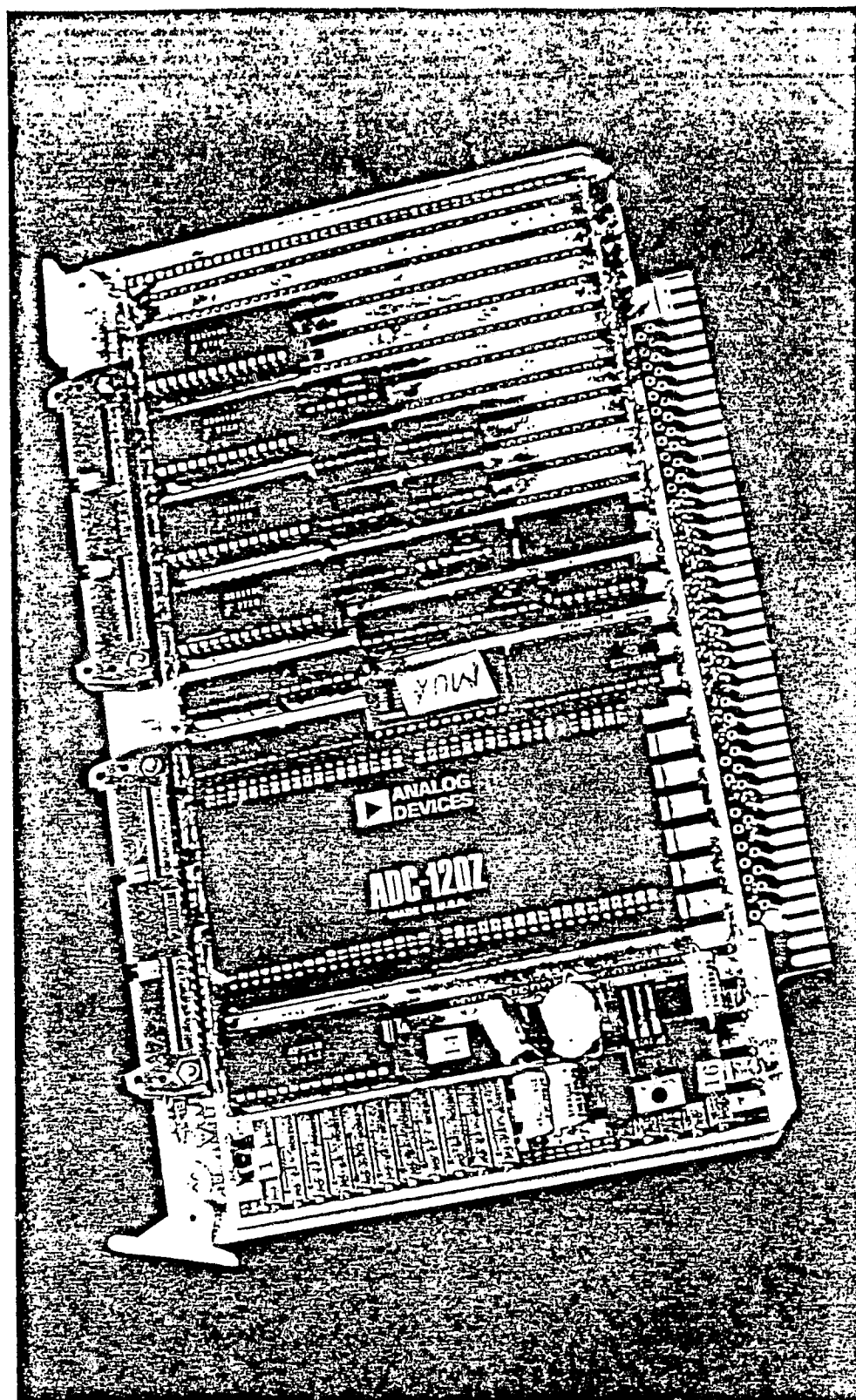
PROM's used are Signetics 8223.

Appendix 6. PROM table for receiver



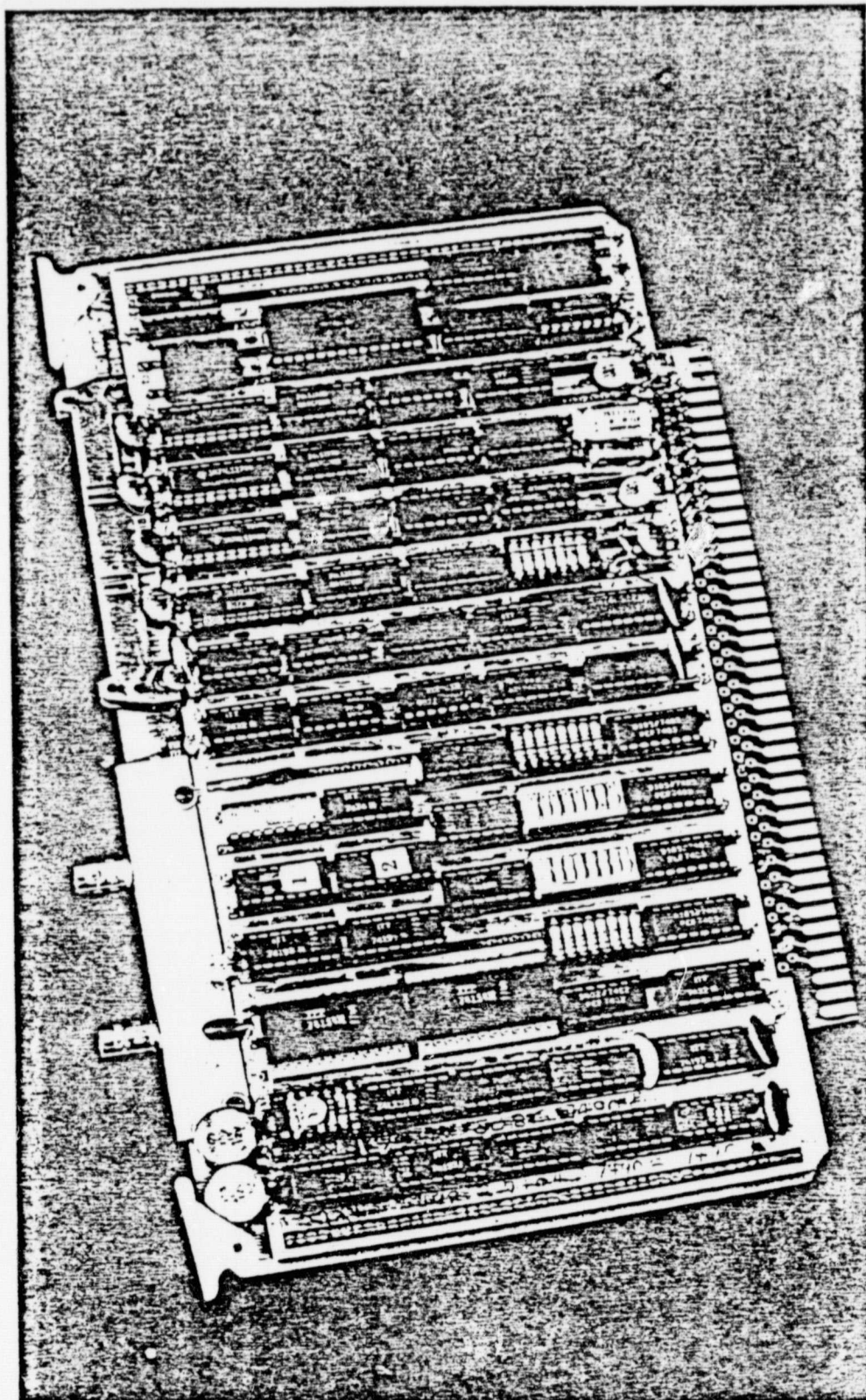
Appendix 7.1. Telemetry Transmitter Board

ORIGINAL PAGE IS
OF POOR QUALITY



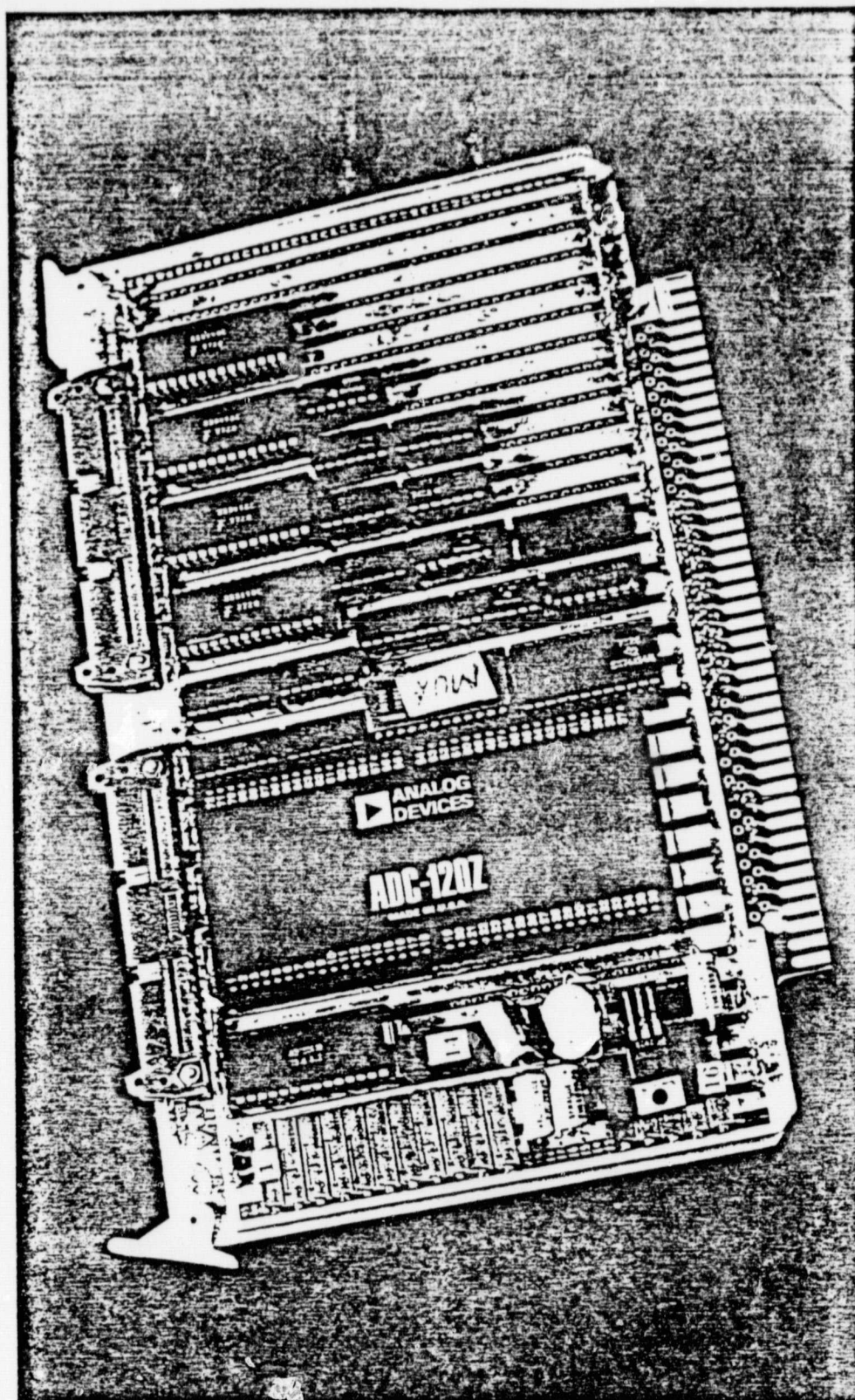
Appendix 7.2. Vehicle Multiplexer Board

ORIGINAL PAGE IS
OF POOR QUALITY



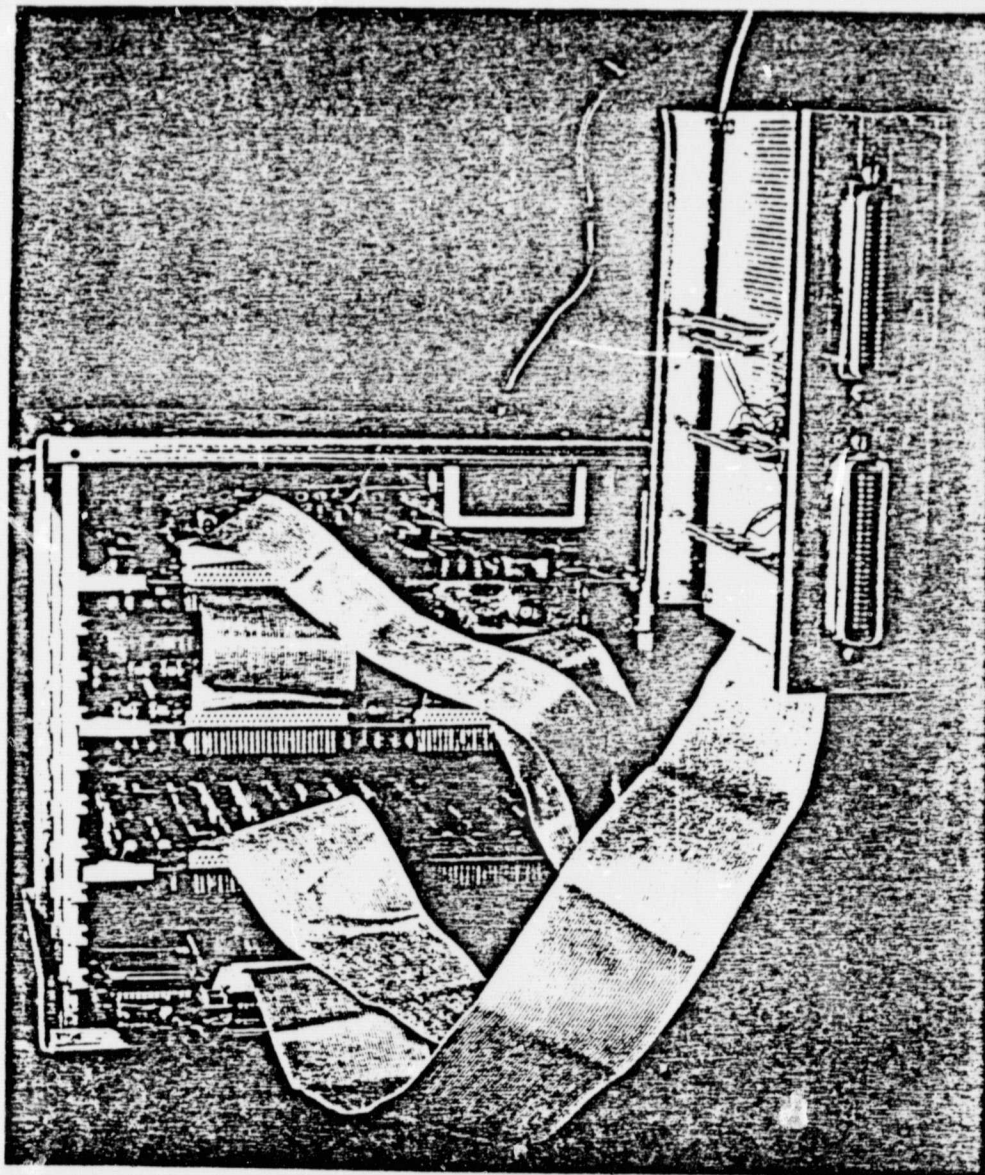
Appendix 7.1. Telemetry Transmitter Board

ORIGINAL PAGE IS
OF POOR QUALITY

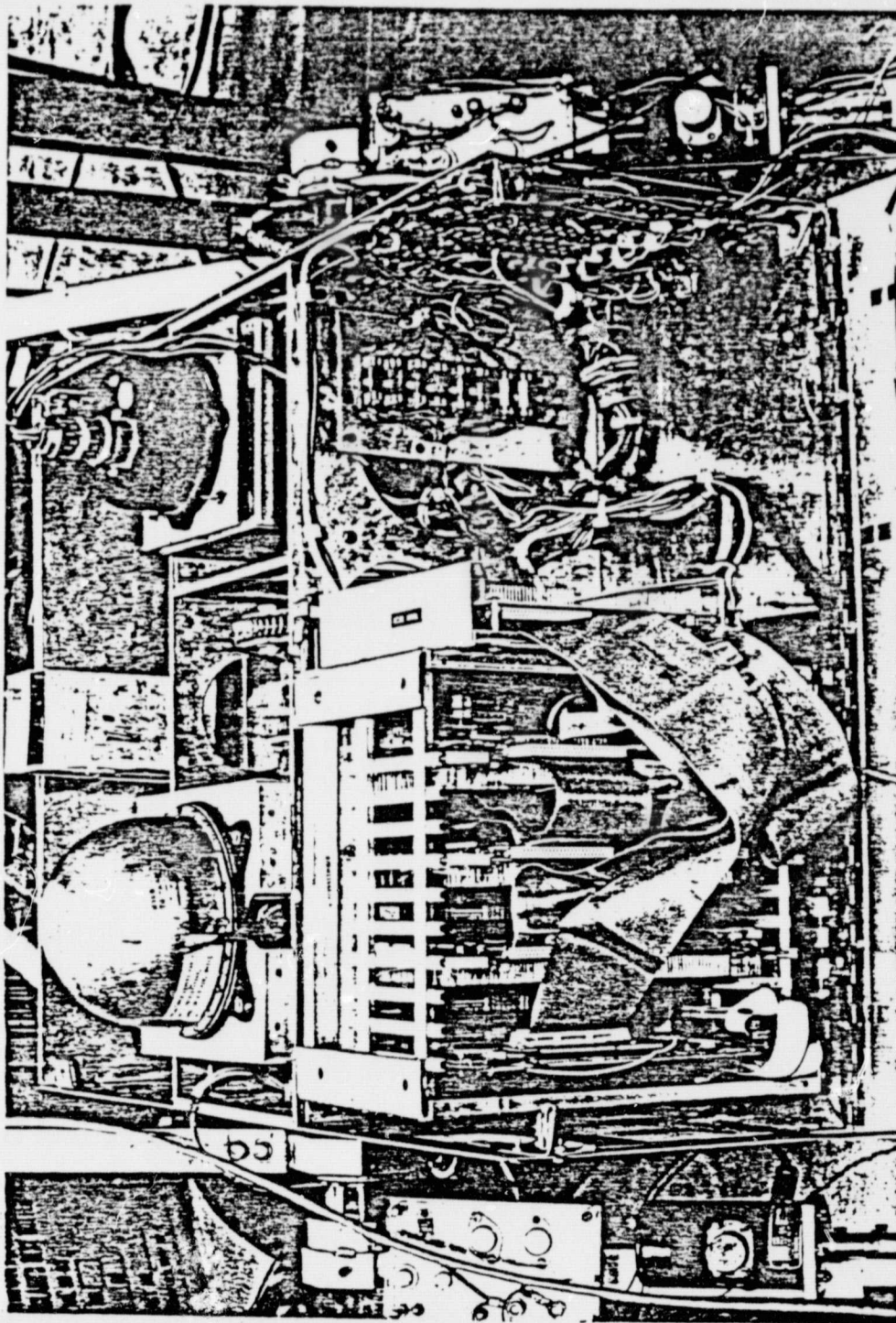


Appendix 7.2. Vehicle Multiplexer Board

ORIGINAL PAGE IS
OF POOR QUALITY

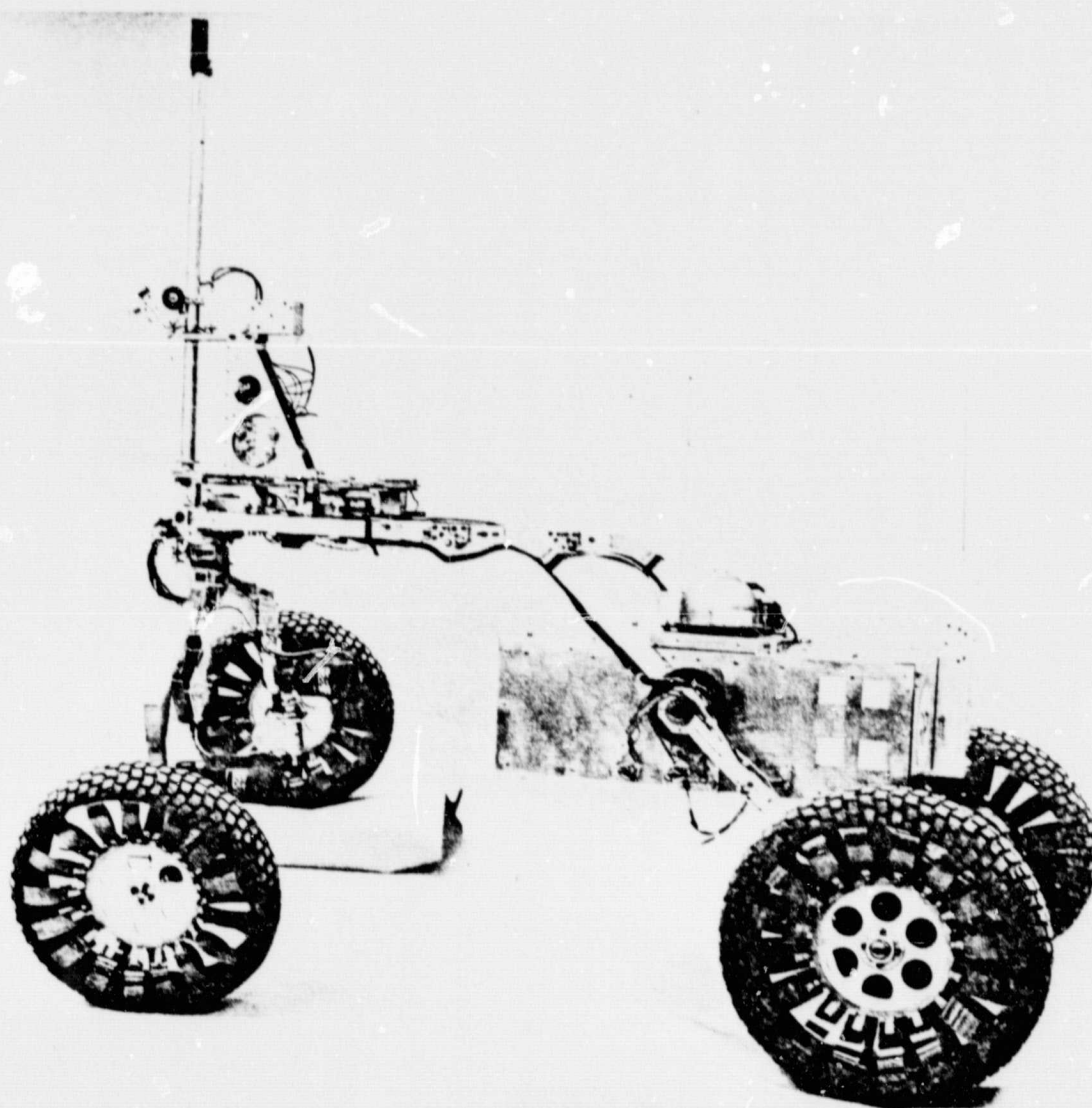


Appendix 7.3. Interconnecting Box (foreground) and Card Cage (background)



Appendix 7.4. Card cage mounted in the autonomous roving vehicle.

ORIGINAL PAGE IS
OF POOR QUALITY



RENSSELAER AUTONOMOUS ROVING VEHICLE

ORIGINAL PAGE 4
OF POOR QUALITY

```

00001      NAM      XMITCD
00002
00003      *      This program is the test program for the telemetry
00004      *      transmitter. It will cause the telemetry board to
00005      *      send data to all possible addresses in the buffers
00006      *      which the receiver fills in the Prime.
00007      *
00008      *
00009      *      The program functions by putting data into the transmitter
00010      *      data latches from the PIAs attached to the micro. The data
00011      *      transmitted for each address is the same as that address.
00012      *      Locations $8006 and $8004 are where the hi and lo data bytes
00013      *      should be stored, respectively, while locations $8022 and $8020
00014      *      are where the hi and lo address bytes should be stored.
00015      *
00016      *
00017      ORG      $00
00018      START  4F
00019      LDX    CE 8004
00020      STA    A $01,X
00021      STA    A $03,X
00022      STA    A $1D,X
00023      STA    A $1F,X
00024      LDA    A $FF
00025      STA    A $00,X
00026      STA    A $02,X
00027      STA    A $1C,X
00028      STA    A $1E,X
00029      LDA    A $04
00030      STA    A $01,X
00031      STA    A $03,X
00032      STA    A $1D,X
00033      STA    A $1F,X
00034      ACAIN  CLR B
00035      CLR    A
00036      BSR    SEND
00037      ADD    B $01
00038      ADC    A $00
00039      CMP    A $07
00040      BNE    HERE
00041      CMP    B $FF
00042      BEQ    EOS
00043      PSH    B
00044      AND    B $3F

                                CLEAR PIA CONTROL REGISTERS.

                                SET DATA DIRECTION REGISTERS TO OUTPUT.

                                POINT TO DATA REGISTERS.

                                CLEAR LO BYTE.
                                CLEAR HI BYTE.
                                SEND ADDRESS AND DATA BYTES.
                                INCREMENT LO BYTE.
                                ADD CARRY BIT TO HI BYTE.
                                IS THIS THE END-OF-SCAN ?

                                YES, THEN SEND END-OF-SCAN INTERRUPT.
                                SAVE ACCUMULATOR B.
                                MASK OFF TOP TWO BITS.

```

00045	0033	C1 1F	CMP B	#31F	IS THIS THE END OF A BLOCK OF AZIMUTH DATA ?
00046	0035	27 10	BEQ	EOV	YES, THEN SEND END-OF-VEHICLE INTERRUPT.
00047	0037	C1 3F	CMP B	#33F	IS THIS THE END OF A BLOCK OF VEHICLE DATA ?
00048	0039	27 03	BEQ	EOA	YES, THEN SEND END-OF-AZIMUTH INTERRUPT.
00049	003B	33 1	PUL B		RESTORE ACCUMULATOR B.
00050	003C	20 E4	BRA	LOOP1	SEND NEXT WORD.
00051	003E	8A 20	ORA A	#20	PREPARE TO SEND END-OF-AZIMUTH INTERRUPT.
00052	0040	33	PUL B		RESTORE ACCUMULATOR B.
00053	0041	8D 0E	CONT1	SEND	SEND ADDRESS AND DATA BYTES.
00054	0043	84 1F	AND A	#31F	RESET EOA, EOS, AND EOVD INTERRUPT BITS.
00055	0045	20 DD	BRA	LOOP2	SEND NEXT WORD.
00056	0047	8A 00	ORA A	#30	PREPARE TO SEND END-OF-VEHICLE INTERRUPT.
00057	0049	20 F5	BRA	CONT1	SEND EOV INTERRUPT.
00058	004B	8A 60	ORA A	#360	PREPARE TO SEND END-OF-SCAN INTERRUPT.
00059	004D	8D 02	BSR	SEND	SEND EOS INTERRUPT.
00060	004F	20 CF	BRA	AGAIN	SEND NEXT BLOCK OF 2K WORDS.
00061					
00062					
00063					
00064					
00065					
00066					
00067	0051	36	SEND		
00068	0052	A6 00	LDA A	\$00,X	SAVE ACCUMULATOR A.
00069	0054	A6 01	SENDLP	\$01,X	CLEAR PIA INTERRUPT BITS.
00070	0056	48	ASL A		CHECK PIA INTERRUPT BITS. IF BIT 6 IS HI,
00071	0057	2A FB	BPL	SENDLP	THEN THE "SO VEH" SIGNAL IS BEING SENT BY
00072	0059	32	PUL A		THE TRANSMITTER.
00073	005A	A7 02	STA A	\$02,X	RESTORE ACCUMULATOR A.
00074	005C	E7 00	STA B	\$00,X	STORE HI DATA.
00075	005E	A7 1E	STA A	\$1E,X	STORE LO DATA.
00076	0060	E7 1C	STA B	\$1C,X	STORE HI ADDRESS.
00077	0062	39	RTS		RETURN TO MAIN PROGRAM.
00078			END		XMITCD

* * * This is the subroutine which puts the address and data bytes
* * * into the proper PIA locations.

SYMBOL TABLE

START	0000	AGAIN	0020	LOOP1	0032	LOOP2	0024	HERE	0030	EOA	003E	CONT1	0040
CONT2	0041	EOV	0047	EOS	004B	SEND	0051	SENDLP	0054				

APPENDIX 9

GLOSSARY OF TERMS

A/D	Short for Analog to Digital converter.
ADLC	Advanced Data Link Controller. In this case, this is a Motorola M68A54 chip which performs the functions of serializing and de-serializing data, error checking, and formatting of the bit stream.
ASK	Amplitude Shift Keying, or PAM with a digital modulating signal.
BYTESEL	Short for byte select. This is a signal which increments a counter to keep track of which data byte is being sent.
EOA, EOS, EOV	Interrupt signals, see Appendix 1.
ECLK	Short for enable clock. This is equivalent to phase 2 (ϕ_2) and is used by the ADLC to clock data through its registers.
FIFO	First-In, First-Out memory. This device is rate buffer which can be used as an "elastic" memory between two systems operating at different speeds.
GPIB	General Purpose Interface Board. In this case, a circuit card used on a Prime computer to enter data into the computer in a direct memory access mode.
ICOM	A General Electric tradename for Internally Compensated Oscillator Module, this device has active internal components to generate a crystal-controlled frequency, fixed over temperature.
INIT	Short for Initial. This is a mode after MR, during which the ADLC gets initialized.
MR	Short for Master Reset. This is a signal issued from a switch on the microprocessor or telemetry board which starts all logic at their first state.
MUX	Short for multiplexer.
PAM	Pulsed Amplitude Modulation. A means of superimposing information on a carrier by varying its amplitude in a pulsed or discontinuous fashion. A form of PAM using only digital levels of carrier is called ASK.

PRIME The name of the computer company which manufactures the computer used for real-time control of the rover. This word is used synonymously here to mean "computer."

PROM Programmable Read-Only Memory.

RAM Random Access Memory.

RDSR Receiver Data Service Request. A signal issued by the ADLC (see ADLC spec).

RF Radio-frequency. Used here to refer to the wireless, carrier link.

RxC Receiver Clock used by the ADLC.

RxD Receiver Data. The serial bit stream recovered for the ADLC.

TDSR Transmitter Data Service Request. A signal issued by the ADLC.

TxC Transmitter Clock. Clock which shifts out data from the ADLC in the transmitter.

TxD Transmitter Data. Data from the transmitting ADLC in a non-return to zero format.