TECHNICAL ADVISEMENT MEMORANDUM NO. 171-3

ONBOARD CHECKOUT AND DATA MANAGEMENT SYSTEM: DESIGN SUPPLEMENT

PRC D-1403

31 March 1967

Prepared for

National Aeronautics & Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama

PLANNING RESEARCH CORPORATION
LOS ANGELES, CALIF. WASHINGTON, D.C.

This document was prepared by Planning Research Corporation under Contract No. NAS8-20367, "An Airborne Evaluating Equipment Study," for the George C. Marshall Space Flight Center of the National Aeronautics & Space Administration. The work was administered under the technical direction of Quality and Reliability Assurance Laboratory, Marshall Space Flight Center, with Walter T. Mitchell acting as project manager.

PREFACE

This memorandum extends and clarifies certain equipment design concepts of the Onboard Checkout and Data Management System. The initial design is documented in an earlier PRC report entitled, Onboard Checkout and Data Management System: Hardware Definition, PRC D-1336, 11 November 1966.

This memorandum is divided into three sections. The first part is intended to convey to the OCDMS design implementer a representative cross section of the logic involved in the Computer Interface Unit (CIU), and external devices such as the Signal Adapters and Control and Display Unit (CDU). The second part, which is necessary background information for software development, is a brief description of the data acquisition process from a programmer's point of view. This includes a set of input/output instruction formats typical of those which may be employed. Also, timing considerations involved in normal operation are reviewed and the system's self-test capability is explored. The third part is a general examination of the circuit aspects of interunit digital data transmission; it concludes with the development of a recommended transmission method.

FOREWORD

This report was prepared by Planning Research Corporation under Contract Number NAS 8-20367, "An Airborne Evaluating Equipment Study," for the George C. Marshall Space Flight Center of the National Aeronautics and Space Administration. The work was administered under the technical direction of the Quality and Reliability Assurance Laboratory, George C. Marshall Space Flight Center, with Walter T. Mitchell acting as Project Manager.

TABLE OF CONTENTS

LIST OF EXHIBITS

I. SYSTEM LOGIC AND INFORMATION FLOW

This section examines each of the major OCDMS units that are controlled by the computer. These are the Computer Interface Unit (CIU), the Signal Adapters, and the Control and Display Unit (CDU). A functional logic design description is given of each unit. The detailed operation of the logic components is discussed with special emphasis on the data transmission involved between units. Sample data and instruction formats are developed to facilitate system understanding.

A. Computer Interface Unit (CIU)

The CIU is shown in Exhibit 1. Lines shown at the left of the diagram are either coming from or going to the computer. Lines shown at the right of the diagram are either coming from or going to external devices (e.g., Signal Adapters, CDU, or PCM). Generally the lines on the left are single connections between integrated circuit line drivers and line receivers (exceptions are noted as "parallel"). The lines on the right are signal pairs terminating in isolation transformers either in the CIU or external devices. No mechanical or packaging arrangement is implied by Exhibit 1.

There are five major functional tasks of the CIU.

1. Information Transfer from Computer to External Device (excluding PCM)
2. Information Transfer from External Device to Computer
3. Internal Information Generation (e.g., Range Time)
4. Information Transfer from Computer to PCM
5. Self-Test

These five tasks are discussed in detail in the following subsections.

1. Information Transfer from Computer to External Device

This operation can be examined as a two-step process: (1) correctly receiving information from the computer, and (2) transmitting that information to the proper external device.

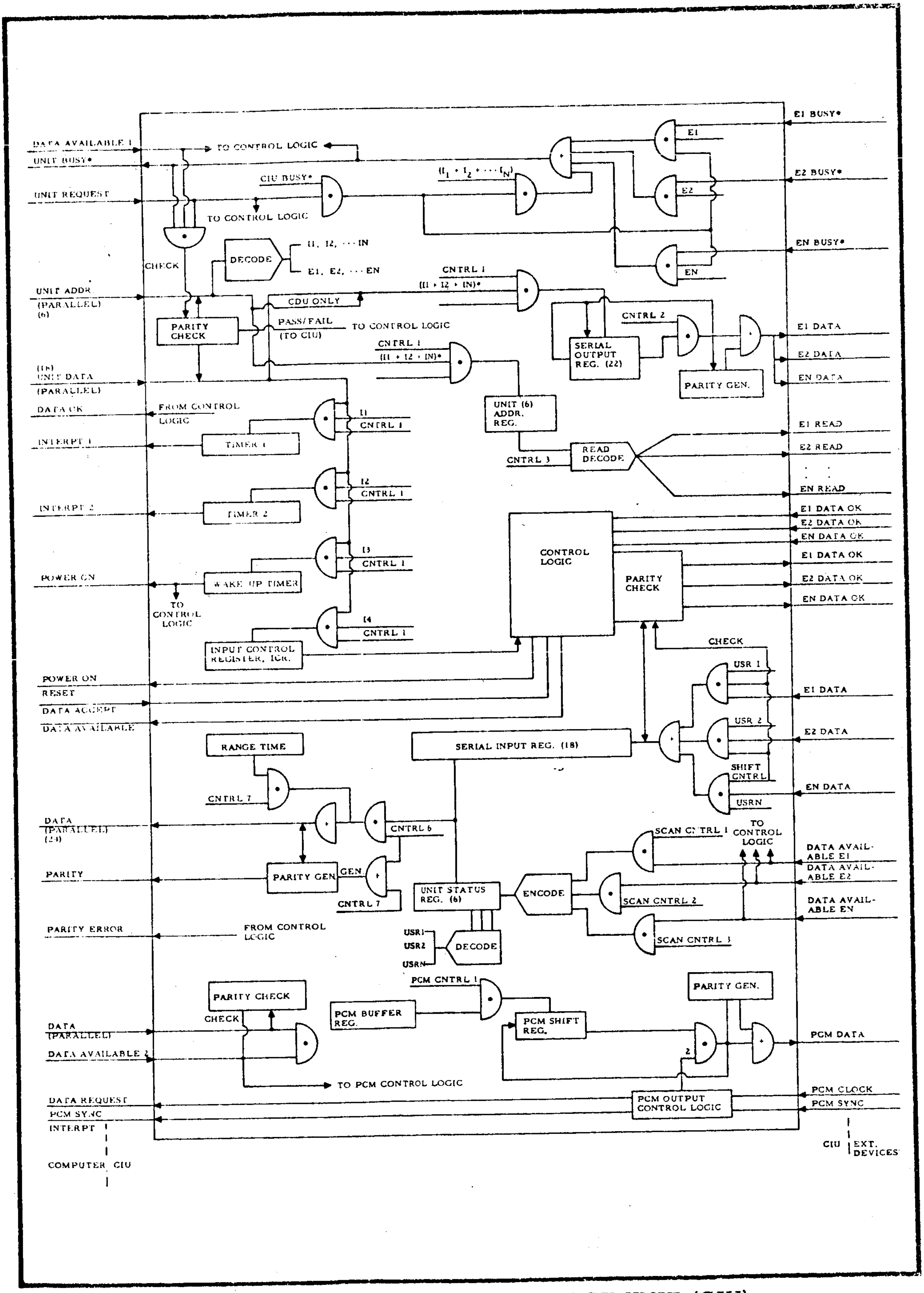


EXHIBIT 1 - COMPUTER INTERFACE UNIT (CIU)

a. Computer to CIU Information Transfer

The basic logic for this operation is shown in the upper part of Exhibit 1. When a computer output instruction is executed, the computer will transmit to the CIU an input/output (I/O) instruction of the form shown in Exhibit 2. Along with this set of 25 lines (24 data, 1 parity), the computer will transmit a data-transfer-request logic signal. In Exhibit 1 this signal is called "Unit Request." The external device code in the I/O instruction is designated as the "Unit Address" in Exhibit 1. The other data and parity bits are designated as "Unit Data" lines.

When a Unit Request is detected, the CIU decodes the Unit Address lines to determine which external device is to receive the data. Exhibit 1 shows two classes of decodes: those for "external devices" internal to the CIU (e.g., Wake-up Timer), and those for devices external to the CIU (e.g., Signal Adapter 1). The former are designated by I1, I2, etc., and the latter by E1, E2, etc. The Ei decodes are compared with the corresponding Not Busy signals from the external devices[1] to determine if the selected device is able to receive information. A Busy/Not Busy signal is then returned to the computer to indicate whether or not the transfer is possible.

If the external device is not busy, the CIU checks for correct parity and transfers the information to the Serial Output Register. The Unit Address is simultaneously transferred to the Unit Address Register. There it is used to direct the subsequent data transfer to the proper external unit. (A minor deviation occurs when the CDU is selected as the external device. It has a special format requiring four of the Unit Address lines to be regarded as data, and, hence, these signals are input to the Serial Output Register along with the 18 data signals.) In the event that the terminal device is within the CIU, then the "Ii" decodes direct the Unit Data signals to the selected internal register instead of to the Serial Output Register. A "Data OK" signal is transmitted to the computer when parity checks out correctly.

[1] "Busy" designations in Exhibit 1 mean Not Busy.

EXHIBIT 2 - INPUT/OUTPUT INSTRUCTION FORMATS

General Format (24-Bit Instruction)

6 Bits	18 Bits
External Device	Device Dependent Format (shown below)

External Device	6-Bit Field	Function	18-Bit Field
Computer Interface Unit	0	Set Wake-up Timer	Time Value (in seconds)
	1	Set Timer 1	Time Value (in milliseconds)
	2	Set Timer 2	Time Value (in milliseconds)
	3	Set Input Control Register	0 - Input Wake-up Timer Value
			1 - Input Timer 1 Value
			2 - Input Timer 2 Value
			3 - Input Serial Output Register Value
			4 - Input PCM Buffer Register Value
			5 - Input PCM Shift Register Value
			6 - Input Range Time Value
			7 - Input Unit Address Register Value
			8 - Input Input Control Register Value
			9 - Set All Ones to Serial Input Register and Unit Status Register and Inhibit Use
			10 - Same as (9) Except Set to All Zeros
			11 - Input Serial Input Register Value and Unit Status Register Value and Remove Inhibit
			12 - Not Assigned on

EXHIBIT 2 (Continued)

External Device	6-Bit Field	Function	18-Bit Field
	4 -15	Not Assigned	
Signal Adapter 1	16	General	Shown on Exhibit 9
Signal Adapter 2	17	General	Shown on Exhibit 9
.	.	.	.
.	.	.	.
.	.	.	.
Signal Adapter 15	31	General	Shown on Exhibit 9
Control and Display Unit	32-47	General CDU	For the CDU this field is 22 bits in length. (See Exhibit 12.)
	48-63	CDU Printer	

The control logic for this operation, which is not explicitly shown in Exhibit 1, resides in the large block designated "Control Logic." It consists of sequencing logic for the control signals, CNTRL 1, 2, and 3, and shift timing and counting. Exhibit 3 shows the important timing relationships involved. The time slots shown should correspond to the basic timing of the selected computer.

An alternative to the busy scheme shown.is routing the external unit Not Busy signals directly to the computer and letting it perform the necessary comparisons prior to initiating a transfer. The drawbacks to such an approach are more lines between the computer and CIU and the difficulty in incorporating this logic into an "off-the-shelf" computer.

b. CIU to External Device Data Transfer

Once the information has reached the Serial Output Register and the Unit Address Register, logic is initiated to transmit the information to the selected external unit. The Unit Address Register is decoded to determine the external unit and a "Read" signal is sent to that unit. The timing is shown in Exhibit 5. This Read signal is a pulse two clock intervals long and informs the external unit that a data transfer is underway. Using the Read signal as a sync pulse, the external unit begins clocking the serial data into its receiving register and simultaneously checking parity. When the proper number of bits have been received and parity checked, a Data OK signal is returned to the CIU.

If the CIU does not receive a Data OK as expected, the Read pulse is again transmitted and the cycle repeated. Because this retransmission may be required, the data in the Serial Output Register are not lost as they are shifted out for transmission to the external unit. Instead it is "rotated" to the beginning of the register so that the original contents of the register will be present after each full register transmission. There are two rotate points depending on whether the full 22 bits are used (as in a CDU transmission), or the normal 18 bits are used (as in Signal Adapter transmission). If the CIU does not receive a Data OK after three transmissions of the same data, it stops trying and informs the computer of its inability to successfully transmit information to that unit.

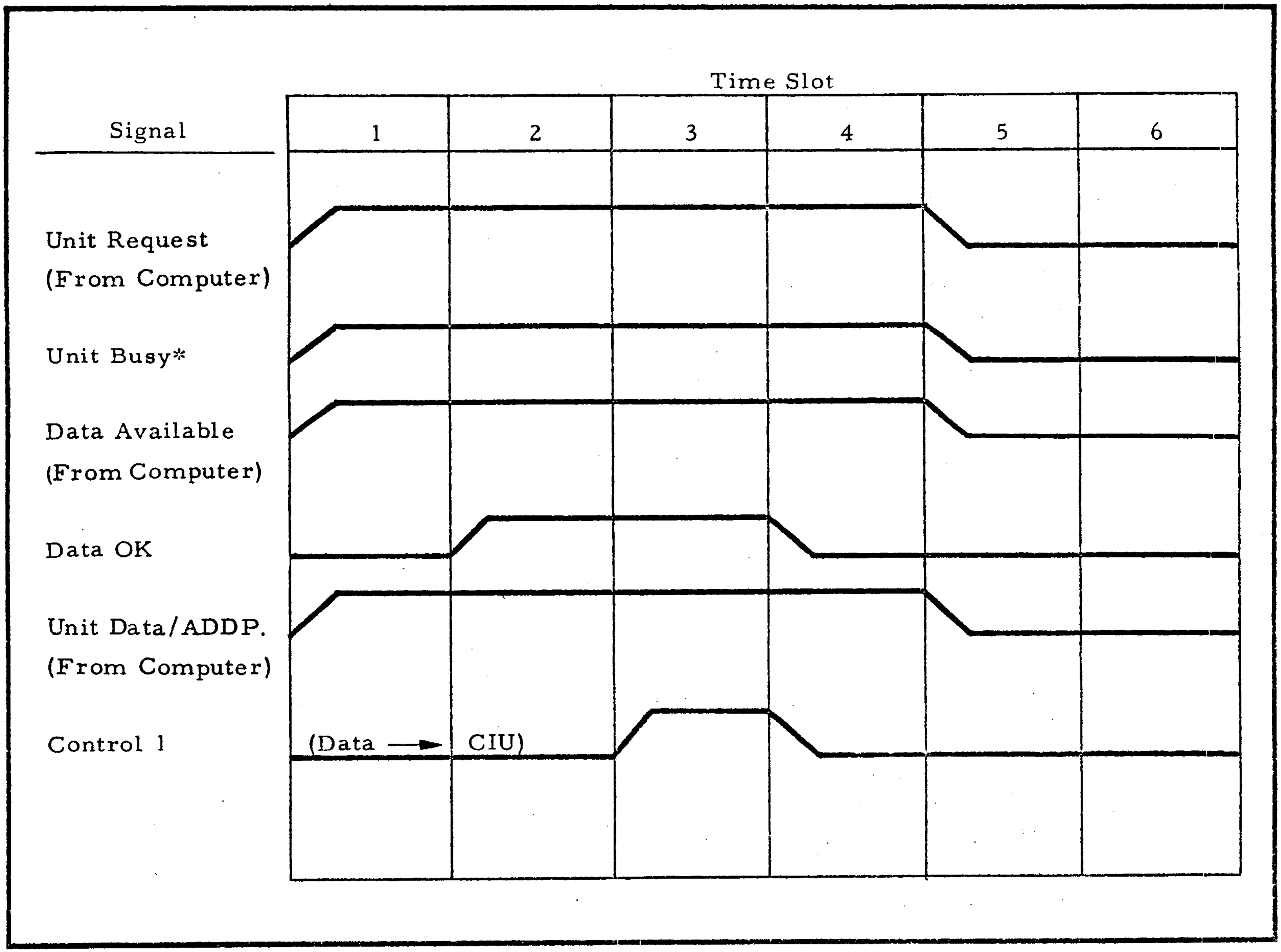


EXHIBIT 3 - COMPUTER TO CIU DATA TRANSFER.

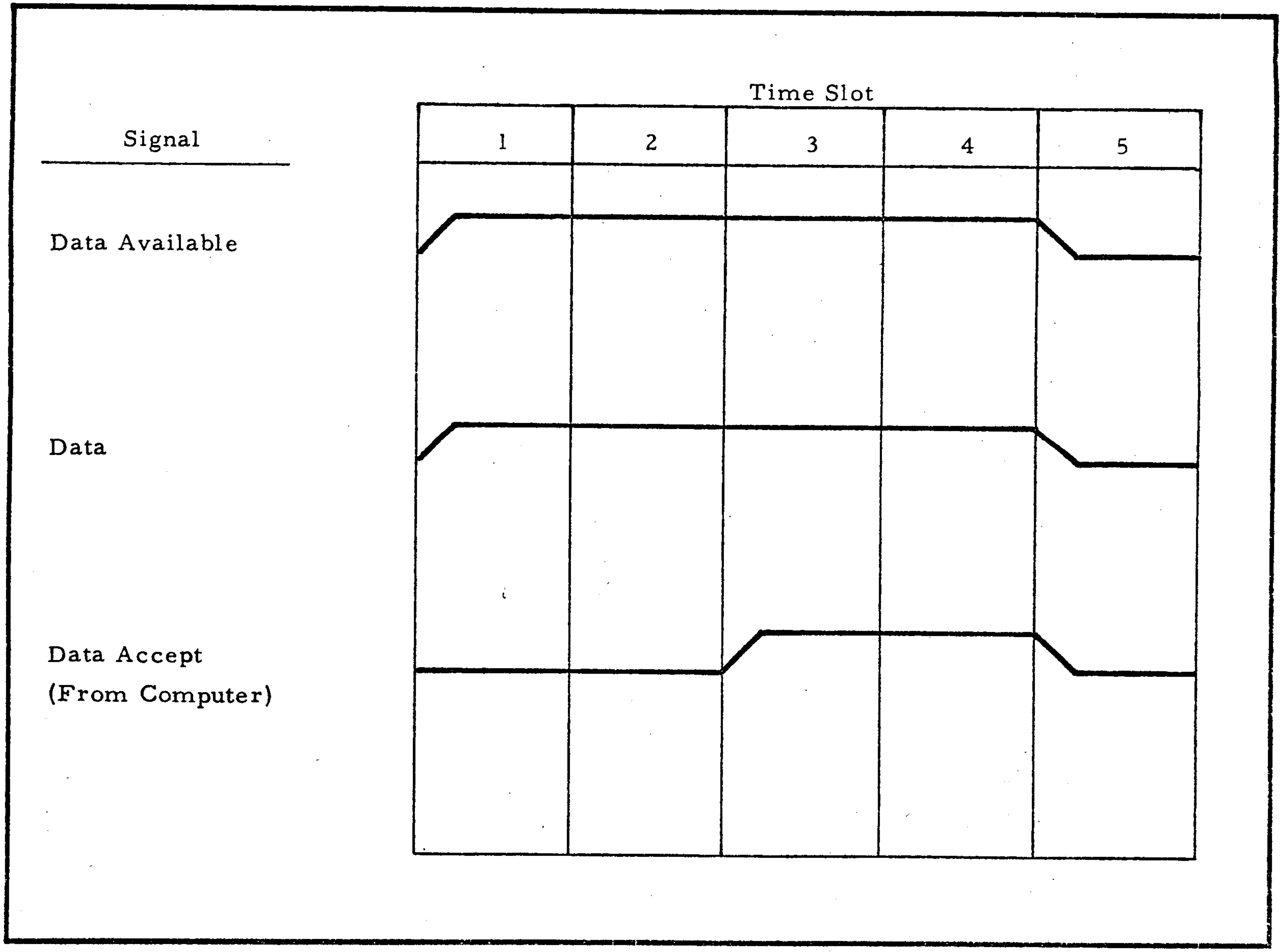


EXHIBIT 4 - CIU TO COMPUTER DATA TRANSFER

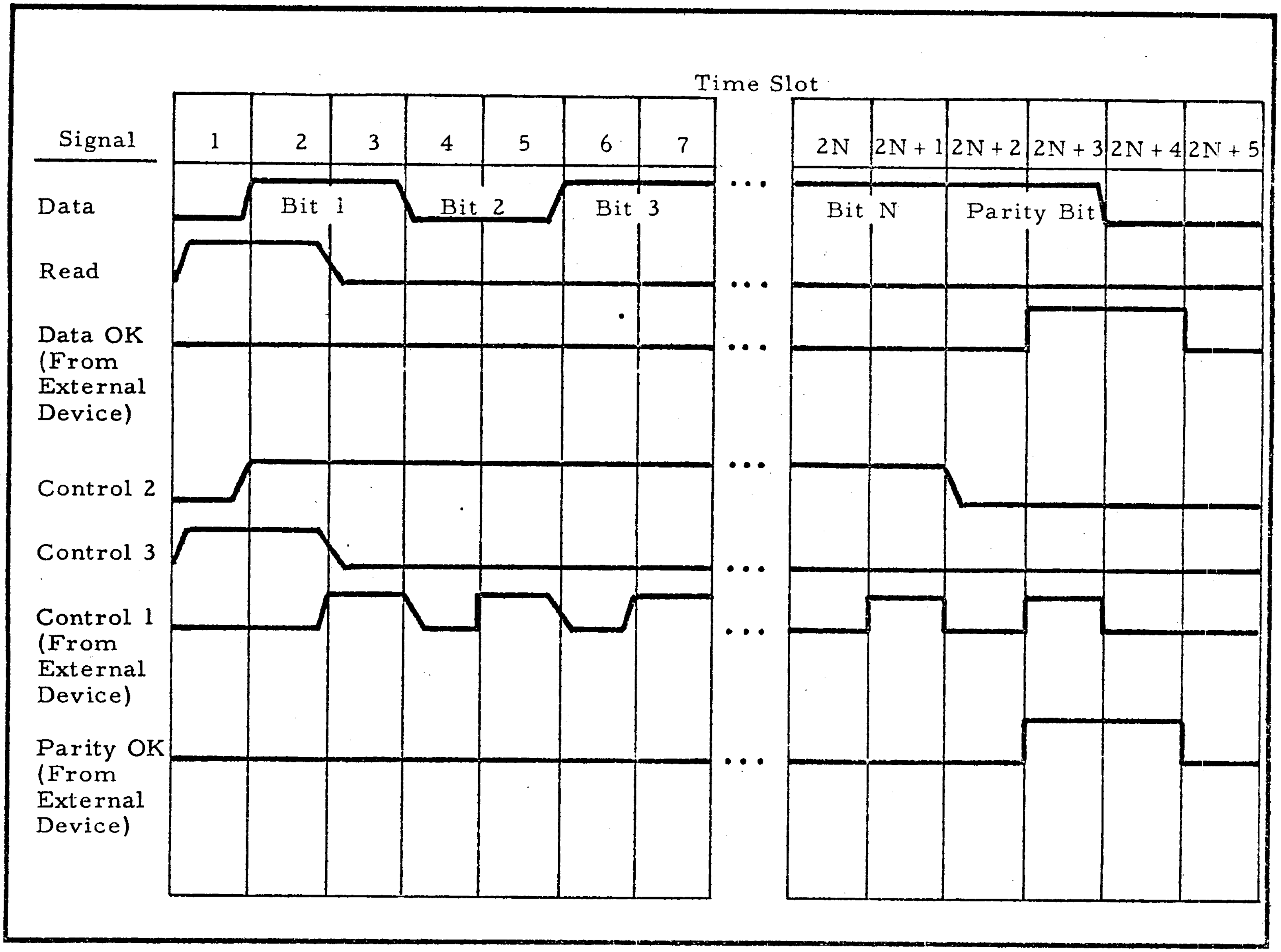


EXHIBIT 5 - CIU TO EXTERNAL DEVICE DATA TRANSFER

2. Information Transfer from External Device to Computer

This operation also can be viewed as a two-step process: (1) transmitting information from the external device to the CIU, and (2) sending the information from the CIU to the computer.

a. External Device to CIU Information Transfer

When an external device has information ready for transmittal to the CIU it sends a "Data Available" pulse to the CIU (see Exhibit 6). This informs the CIU that data is being transmitted to it from the external device. Since the CIU receives information from many external devices, it has a priority scheme to select which device it will receive in the event of simultaneous Data Available signals from two or more external devices.

If the CIU is free and no higher priority device is simultaneously asking for attention, the CIU clocks the serial data stream into the Serial Input Register and checks parity. If the parity check indicates a successful data transmission, a Data OK signal is sent to the external device terminating its transmission. Until it receives a Data Available signal, the external device will continue to cyclically transmit the data. If the CIU fails to successfully receive the data after three tries, it so informs the computer and henceforth ignores that external device (or ignores it until instructed not to receive the data by the computer).

b. CIU to Computer Information Transfer

Once the data has been received from the external device and is contained in the Serial Input Register, the subsequent transfer to the computer begins. The contents of the Serial Input Register and of the Unit Status Register (which indicate the device associated with the data) are sent in parallel to the computer. Data Available and parity signals accompany the data. These lines remain static until either a "Data Accepted" signal is returned from the computer or the computer resets the CIU via a command in the Input Control Register (see Exhibit 4). The general format for this information is shown in Exhibit 7.

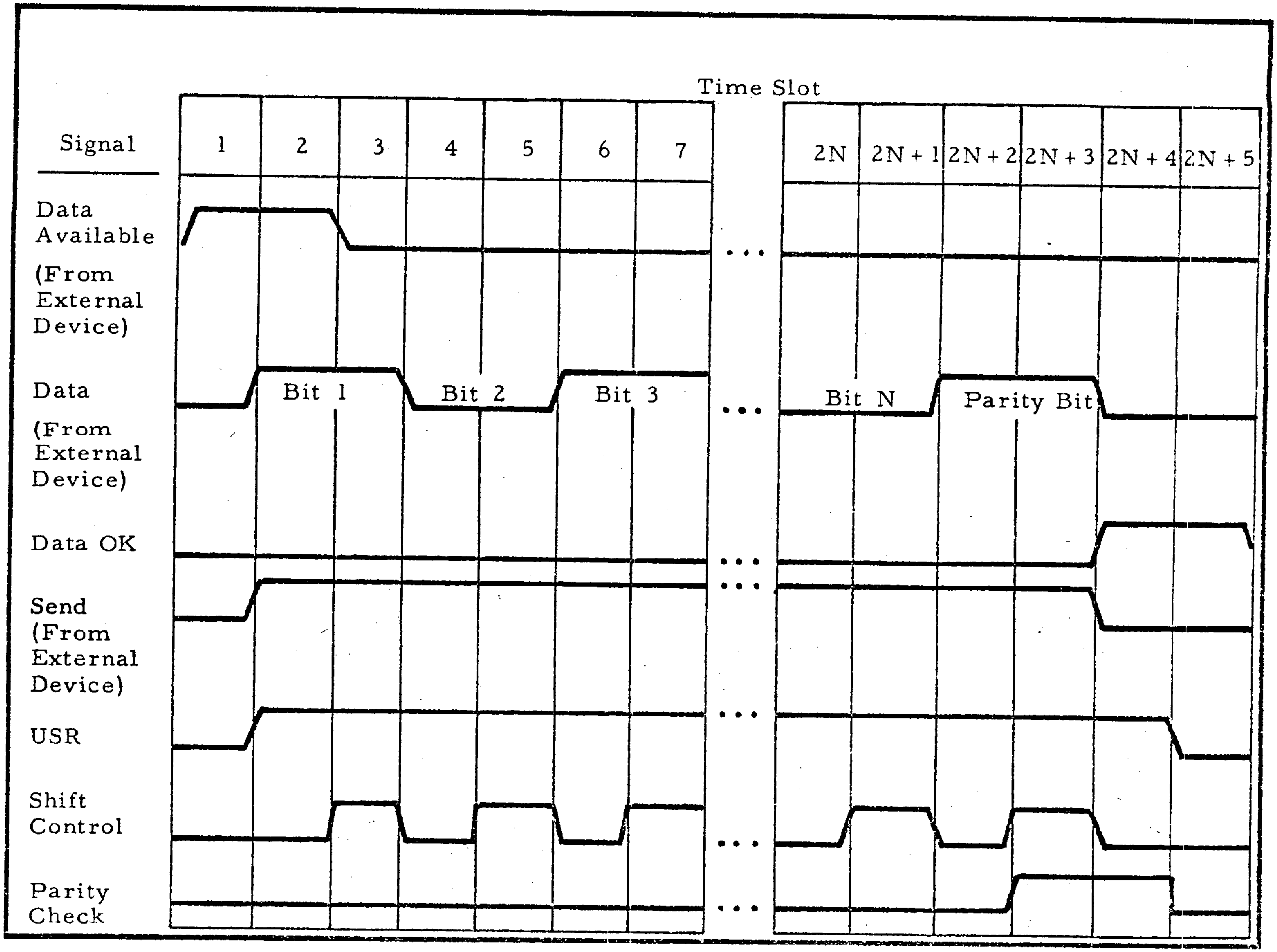


EXHIBIT 6 - EXTERNAL DEVICE TO CIU DATA TRANSFER

EXHIBIT 7 - DATA FORMATS

General Format

6 Bits	18 Bits
External Device	Data

External Device		Data*
CIU: Wake-up Timer	0	Value (in seconds)
CIU: Timer 1	1	Value (in milliseconds)
CIU: Timer 2	2	Value (in milliseconds)
CIU: Input Control Register	3	Contents (4-5 bits)
CIU: PCM Buffer Register	4	Contents
CIU: PCM Shift Register	5	Contents
CIU: Range Time	6	Value (in 100 μsec units)
CIU: Unit Address	7	Contents (6 bits)
CIU: Serial Output Register	8	Contents (18 bits)
CIU: Unit Status Register and Serial Input Register	9	Contents (6 bits)
Not Assigned	10-15	
Signal Adapter 1	16	See Exhibit 12
	.	
	.	
	.	
Signal Adapter 16	31	
Control and Display Unit	32-63	Self Test (8 bits register code, 14 bits contents) Input all zeros and 7 bit key code

3. Internal Information Generation

Included under this heading are the four timing registers in the CIU. These registers (except for the Wake-up Timer) are programmer-convenience devices intended to simplify certain programming tasks.

The Range Time Register is a counter that keeps track of mission time. It can be reset to zero by the computer but not set to any other value. With that exception it is a read-only register.

The two timing registers, Timer 1 and Timer 2, are used to generate interrupts to the program at prescribed moments. This relieves the programmer of some of the time-keeping chores persuant to real-time automatic checkout. Both of these registers countdown from the initial value to zero and then generate an interrupt.

The Wake-up Timer is used to conserve power. This counter is always on in "Active," "Standby," and "Self-Test" system modes. In the Standby mode, the computer will input a number to this timer and then shut off itself and everything else in the system. The exception is the Wake-up Timer which remains on and continues to countdown the number supplied it. When it reaches zero, a "Power On" signal is sent to the Power Supply reactivating the computer. Subsequently, either from the CIU or Power Supply, a "Wake-up" reset signal is sent to the computer.

4. Information Transfer from Computer to PCM

The CIU receives a PCM sync pulse from the PCM equipment which is passed in the form of an interrupt to the computer by the CIU. This pulse is used by the computer to prepare a new frame or subframe of PCM data. The CIU control logic is initiated by the sync pulse and requests data from the computer as required by the PCM equipment. These data are clocked out to the PCM transmitter by a clock from the PCM equipment.

The data from the computer are transmitted over a buffered I/O channel distinct from the one used for the other computer CIU-information transfers. This is necessary because of the relatively high data rate required and low waiting tolerance.

5. Self-Test

During this task, input instructions are sent to the CIU's Input Control Register for execution. These instructions are basically routing commands for input of internal registers to the computer. Thus, the computer can request the contents of any CIU internal register to be routed back to the computer via the route followed by external data. These registers in effect become external devices and are sent to the computer just as the contents of the Serial Input Register were in normal operation (see subsection I.A.2.b).

This relatively simple process permits a high degree of fault isolation when coupled with end-to-end test such as routing the Serial Output Register data into the Serial Input Register (i.e., as if the Serial Output Register was an external device).

In addition to the logic operation of the CIU just discussed, there is another relevant aspect depicting the CIU (shown in Exhibit 1). This is the number of lines into and out of the CIU. The following table is a summary of the minimum requirements as shown in Exhibit 1.

Number of lines to computer	35
Number of lines from computer	40
Number of signal pairs to external devices[1]	3n + 2
Number of signal pairs from external devices	4n + 4

These numbers are minimal since Exhibit 1 does not include lines for clock signals, power, grounding, or spares.

B. Signal Adapter

The Signal Adapter shown in Exhibit 8 contains four standard modules: Analog Stimulus, Discrete Stimulus, Discrete Measurement, and Analog Measurement. A single Signal Adapter unit could actually contain as many as eight modules instead of just the four shown. However, the logic shown is general and can be extended to any number of modules.

The left-hand side containing the Receiving Register and the Transmitting Register is common to all Signal Adapter units. This section

[1] The number of external devices excluding PCM and digital command receiver is designated by n.

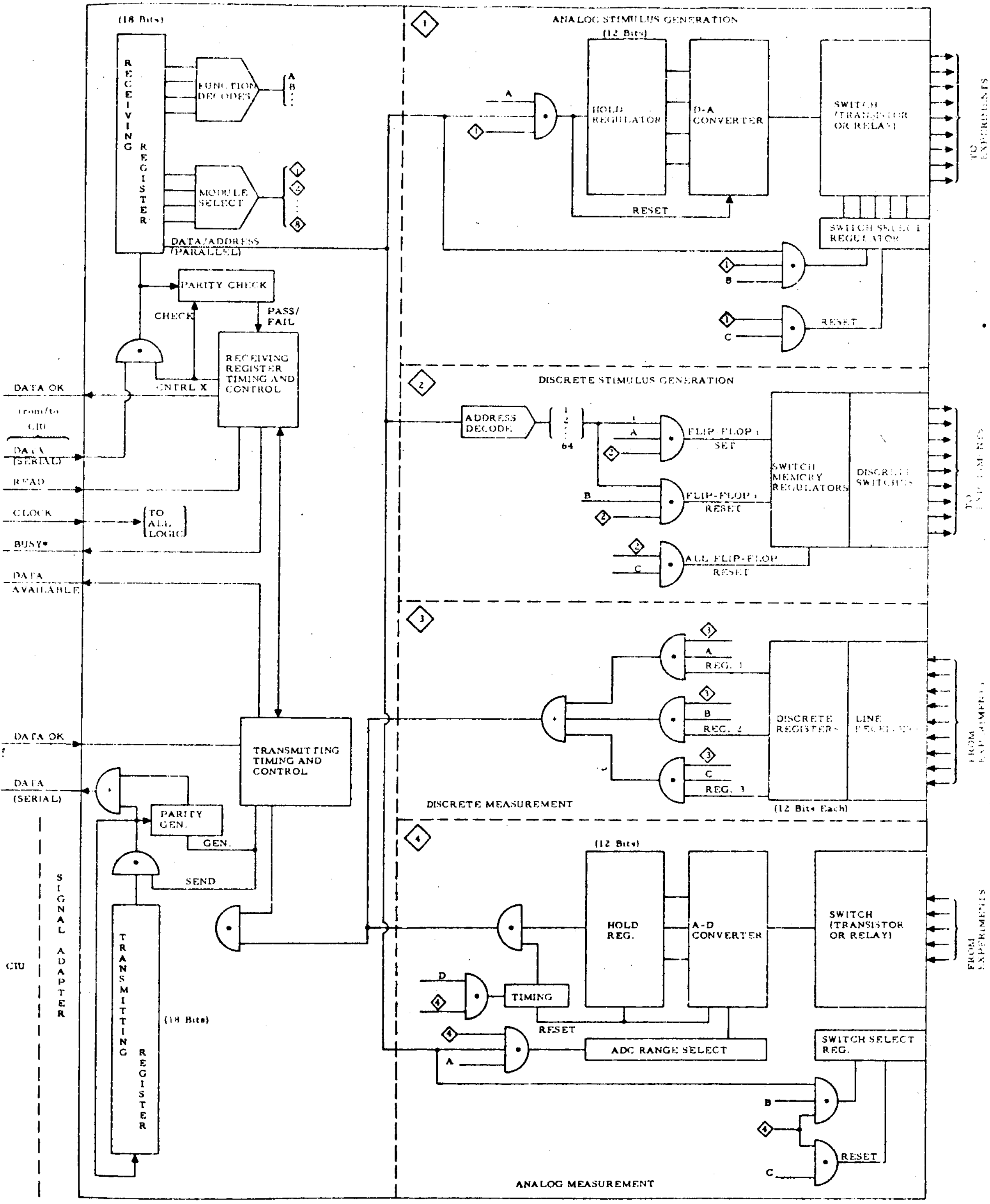


EXHIBIT 8 - SIGNAL ADAPTER

contains the necessary logic for two-way communication with the CIU. It provides a standard set of control functions for operating the various modules. Each instruction received by the Signal Adapter contains module identification and function selection information. These information fields are decoded and made available to each module position. Each module recognizes its own identification decode and obeys the function decodes in a characteristic manner. This standardization will be discussed in more detail in the following subsections.

1. CIU to Signal Adapter Information Transfer

This information transfer was discussed in subsection I.A.1.b. Once this transmission has begun, the Busy* signal is switched off and the unit cannot receive further information until it has processed the new information.

2. Internal Operation

The Signal Adapter instruction formats are shown in Exhibit 9. The resulting operation for each instruction is discussed in the following subsections.

a. Analog Stimulus Generation

When the full 18-bit instruction has been correctly input to the Receiving Register, the module- and function-select decodes are switched on. In Exhibit 8, the Analog Stimulus module is shown in module position 1, and, hence, the module decode ⟨1⟩ will be true (assuming correct programming).

If a 1 has been coded in the function field, the data value portion of the instruction is routed to the Hold Register and the D-A conversion is initiated. In Exhibit 8, this 1 is indicated by an "A" control signal

If a 2 has been coded in the function field, the data value portion of the instruction is routed to the Switch Select Register to address the proper application point. This 2 decode is shown in Exhibit 8 as a "B" control signal.

If a 4 has been coded in the function field, the switch is cleared. The data value portion of this instruction is not used. This decode is shown as a "C" control signal in Exhibit 8.

EXHIBIT 9 - SIGNAL ADAPTER INSTRUCTION FORMATS

General Format (18-Bit Instruction)

3 Bits	4 Bits	11 Bits
Module Position	Function Selection	Data Value/Switch Address (see below)

Module position field references any of eight-module positions in the signal adapter (0-7)

Module[1]	Position	Function Selection	11-Bit Field Meaning
Analog Stimulus Generation	1	1 (A)[2]	Stimulus Value
		2 (B)	Switch Address
		4 (C)	Not Used (this instruction clears switch)
		8--not assigned	
Discrete Stimulus Generation	2	1 (A)	Discrete Address (discrete "on")
		2 (B)	Discrete Address (discrete "off")
		4 (C)	Not Used (this instruction turns all discretes "off")
		8--not assigned	
Discrete Measurement	3	1 (A)	Not Used (reads first discrete register)
		2 (B)	Not Used (reads second discrete register)
		4 (C)	Not Used (reads third discrete register)
		8	Not Used (reads fourth discrete register)
Analog Measurement	4	1 (A)	ADC Range Selection Value
		2 (B)	Switch Address
		4 (C)	Not Used (this instruction clears switch)
		8 (D)[3]	Not Used (initiates conversion)

Notes: (1) Only those modules shown in Exhibit 8 have been examined.

(2) Letters in parenthesis refer to control signals in Exhibit 8.

(3) As shown, this field is not binary-encoded; hence, more than one function may be selected simultaneously. In this case, functions 1 and 8 might be selected instead of requiring separate commands.

Thus, depending on the function selected, a particular analog signal can be generated, switched to any desired point, or the module can quickly be disconnected from the experiment.

b. Discrete Stimulus Generation

If the module position had been coded as a 2, the Discrete Stimulus module (as shown in Exhibit 8) would have been selected and a different instruction format would be applicable. In this instance, both A and B control decodes would interpret the data value portion of the instruction as an address and use it to select a flip-flop. These flip-flops provide the memory for the discrete switches. If a 1 or A decode had been selected, then the addressed flip-flop would be "set" and the discrete switch "closed." If a 2 or B decode had been selected, then the addressed flip-flop would be reset and the discrete switch opened.

In this module, a 4 or C decode is used to clear or reset all of the flip-flops.

c. Discrete Measurement

For this module, each of the function decodes is used to designate a bank of status flip-flops (12 in each). Depending on the function decode, a particular bank is read into the Transmitting Register with a code indicating which bank and which discrete module. The format for this data is shown in Exhibit 10. The data value field is not used by this module.

d. Analog Measurement

When a 1 or A function is coded, the data value field is sent to the ADC Range Select Register to determine in which range the ADC will take place.

The B decode is interpreted as a switch-selection operation and the data-value field is sent to the Switch Select Register as the address of the measurement point. A C decode will reset the switch.

A D or 8 decode initiates the actual conversion. This may be given simultaneously with an A decode.

EXHIBIT 10 - SIGNAL ADAPTER DATA FORMATS

Analog Measurement (18 Bits)

2 Bits	1 Bit	3 Bits	12 Bits
Type 1	O' Flow	Scale Factor	Measurement Value

Discrete Measurement (18 Bits)

2 Bits	2 Bits	2 Bits	12 Bits
Type 2	Not Used	Reg. No.	12 Unitary Coded Descrete Indicators

Others Not Assigned

When the analog signal has been converted to a digital equivalent, the number is sent to the Transmitting Register in the format shown in Exhibit 10.

e. Other Modules

These same guidelines (as discussed above) are involved in the design of other modules. Each module recognizes a select signal (module position decode) and uses the standard available control signals to implement its operation.

3. External Signal List

There are four twisted pairs from the CIU to the Signal Adapter and four back to the CIU.

The number of lines to the experiments is a function of the maximum number of modules and number of connections per module. If eight modules are permitted in a single Signal Adapter, and if each module has a 64-point switch, then there are 8 x 64 x 2 = 1,024 pins for experiment hookup. However, it may not be practical to permit eight modules in one Signal Adapter or to implement a 64-point switch.

C. Control and Display Unit

The CDU is shown in Exhibit 11. The primary purpose of the logic shown is to control the electroluminescent display characters and to interpret the keyboard inputs. The self-test logic, not shown in Exhibit 11, is discussed in this subsection.

1. CIU Communication

The communication with the CIU is identical to the Signal Adapter-CIU communication with one exception. Each word contains 22 bits of information instead of 18 as is the case with Signal Adapter-CIU communication.

2. Display Control

Exhibit 12 shows the general classes of instructions that may be sent to the CDU. Once received, with correct parity in the Serial

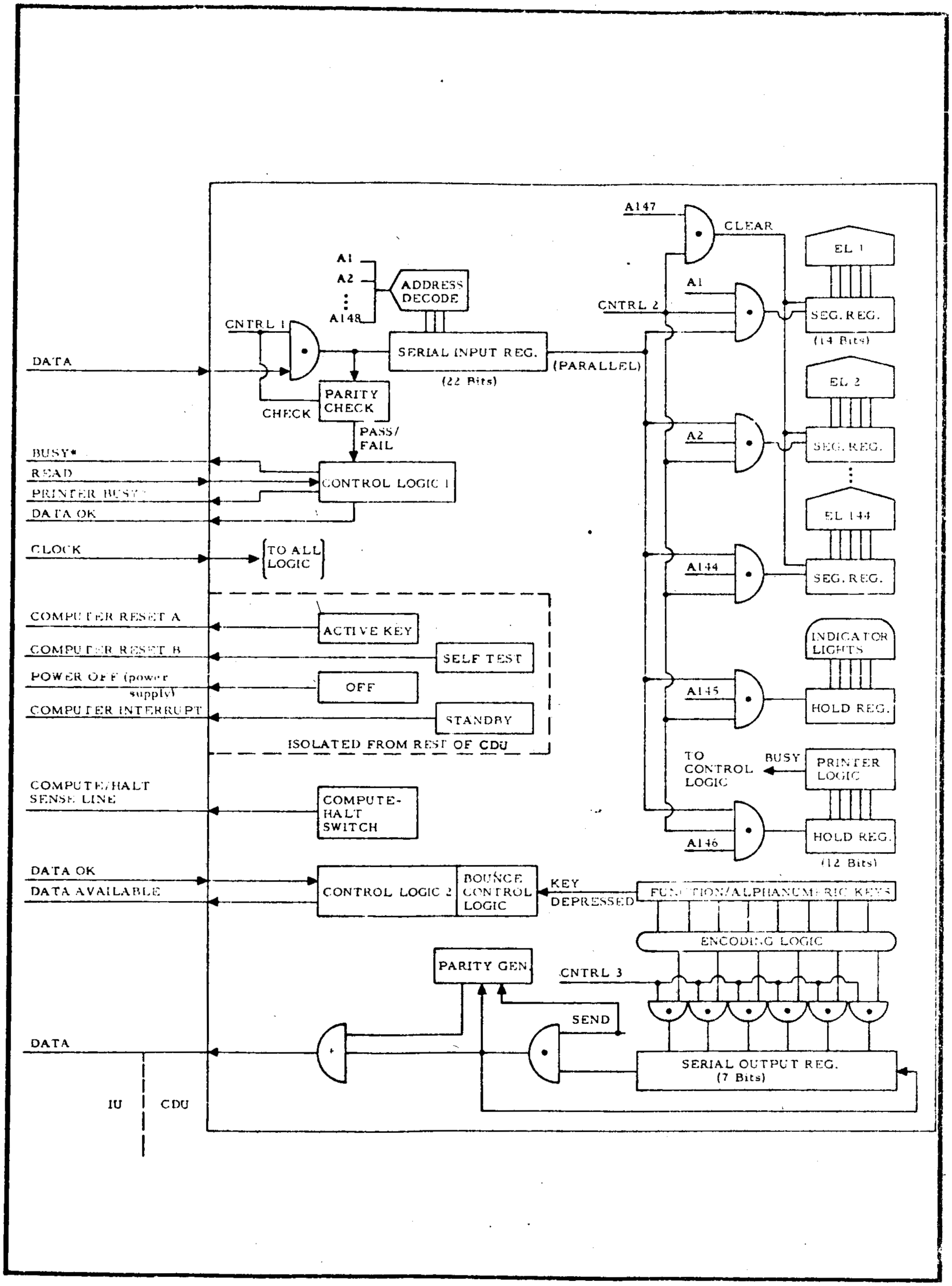


EXHIBIT 11 - CONTROL AND DISPLAY UNIT (CDU)

EXHIBIT 12 - CDU INSTRUCTION FORMATS

General Format (22 Bits)

8 Bits	14 Bits
Destination	Data

Destination	Field	14-Bit Field
Electroluminescent Display Matrix	1	Unitary-Coded Segment On/Off Control
	.	.
	.	.
	.	.
	144	.
	147	Not Used (clears entire matrix)
Indicator Lights	145	Unitary-Coded Segment On/Off Control
		0 - Uplink Message
		1 - on Not Assigned
Strip Printer	146	2 6-Bit Printer Characters
Not Assigned	0	
	148-254	
Self Test	255	CDU Register Code

Input Register, the "Destination" field is decoded and the "Data" field value is routed to the appropriate register. If the Destination field is 255 indicating "self-test," then a slightly different action takes place which is discussed in subsection I.C.6.

These are 144 14-bit registers for the electroluminescent displays. Each register controls one of the alphanumeric electroluminescent characters. Each character has 14 segments associated with it and each bit in the 14-bit register uniquely controls one segment. If a bit is set, the the corresponding segment is energized. If it is reset, then the corresponding segment is off.

One destination code, 147, is used to reset all of the 14-bit electroluminescent registers.

There are several on/off indicator lights on the panel. These are also controlled by flip-flops in a holding register. The exact number of these is a function of final CDU panel design.

A 12-bit holding register for the printer is used to hold two 6-bit alphanumeric codes for printer output. Because the printer action is very slow, relative to the EL and indicator light displays, a separate Busy* line is used to indicate its status. In this way the CDU is not tied up by the printer and the EL's can be changed while the printer is operating.

3. Input Control

The seven bits of data from the Function/Alphanumeric keys provide six bits of function or alphanumeric data and one bit which indicates whether the other six bits should be interpreted as a function code or as an alphanumeric code. This seventh bit is controlled by a two-position switch which eliminates the necessity of depressing it with each entry. The other keys are all momentary-action switches.

4. System Mode Control

The system mode keys are located in the CDU but share grounding and power with the computer rather than the rest of the CDU. The four lines from the mode keys are not isolated from the computer via pulse transformers as are the other signal pairs between the CDU and the CIU. The active key resets the computer to a fixed location and turns on computer power if it is not already on. The off key turns off system power. The self-test key resets the computer to a fixed memory location different from that used by the active reset. The standby key is an interrupt to the computer indicating a desire to enter the standby operational diode.

5. CIU-CDU Signal Lines

There are 10 signal pairs connected to the CIU. Six of these pairs go to the CIU and four of them come from the CIU.

6. CDU Internal Self-Test

The self-test logic was omitted from Exhibit 11 to avoid cluttering the diagram. When a 255 Destination code is decoded in the Serial Input Register, the Data field is then used to specify an internal register whose contents are to be transmitted back to the CIU. For example a Data field value of 23 might designate the 14-bit register for electroluminescent character number 23. The contents of that register plus the register code would then be transmitted to the CIU. In this way the logic can be verified to the flip-flop level, and, coupled with an operator visual check of the panel, a reasonably detailed level of fault isolation can be achieved.

II. PROGRAMMER REFERENCE INFORMATION

This section clarifies two points of interest to the software designers which were not explicitly covered in the OCDMS hardware report (see Foreword). The first point involves the inherent delays in operation due to OCDMS and experiment hardware reaction times that must be taken into account by the software. The second point clarifies the extent to which self-test features are built into OCDMS and the recommended method of utilizing them.

A. Timing Considerations

When sequences of instructions are sent to an external device, the programmer must pay particular attention to delays in instruction execution which are inherent in the OCDMS. (It goes without saying that the programmer must provide for all experiment hardware or operational delays.) As an example of these delays, consider the following sequence of instruction to Signal Adapter 1.

	6 Bits	18 Bits			
1	16	1	1	5	Setup DAC for 5 volts
2	16	1	2	13	Apply to point 13
3	16	4	2	14	Connect ADC to point 14
4	16	4	1	4	Select range 4 (e. g., 1-10 volts)
5	16	4	8	--	Measure point 14

Since each of these instructions initiates a hardware operation, there is some delay in executing each of them. About four time intervals are required to transfer an instruction from the computer to the CIU (see Exhibit 3). Another 40 intervals are required to then transmit an instruction to the Signal Adapter (see Exhibit 5 for N = 18). Hence, there are at least 44 microseconds (using a one-megacycle clock) between instructions at Signal Adapter. This provides the programmer with a buffer and permits him to concern himself only with the hardware actions which are slower than 44 microseconds.

Although the digital-to-analog converter has not been designed in detail, it is well within the state of the art to achieve the conversion within the 44 microseconds. Hence, the programmer will not, in all probability, have to concern himself with a programmed delay after the first instruction since the DAC will be steady at 5 volts by the time his second instruction is executed.

In the sequence shown, the second, third, and fourth instructions can be sent immediately following one another without timing problems. There are three delays to be considered before transmitting the fifth instruction, however. Associated with instructions 2 and 3 are switch closure delay times which will be fixed for each different type of switch. There will be two general classes of switches: solid state and relays. The solid-state switching is very fast and may not require programmer accommodation. Conversely, even the fastest relays are very slow relative to a 44-microsecond buffer time and will require programmers to wait before issuing sequential instructions which employ relays. In addition to the switch delays within the OCDMS, the programmer at this point must allow for delays in response from the experiment hardware. When all delays have been accounted for and the programmer has determined that the desired signal is ready and waiting at the analog-to-digital converter, he may then issue the fifth instruction.[1]

In general, the programmer will have to determine the delay, if any, inherent in each I/O instruction. In this type of system, the programmer must interface closely with the hardware designer in order to accomplish this.

B. Self-Test Considerations

The programmer must also work closely with the hardware designer to ensure proper hardware/software implementation of an adequate self-test scheme. Exhibit 13 shows the primary levels of self-test

[1] Range selection will result in a delay similar to the switch delays.

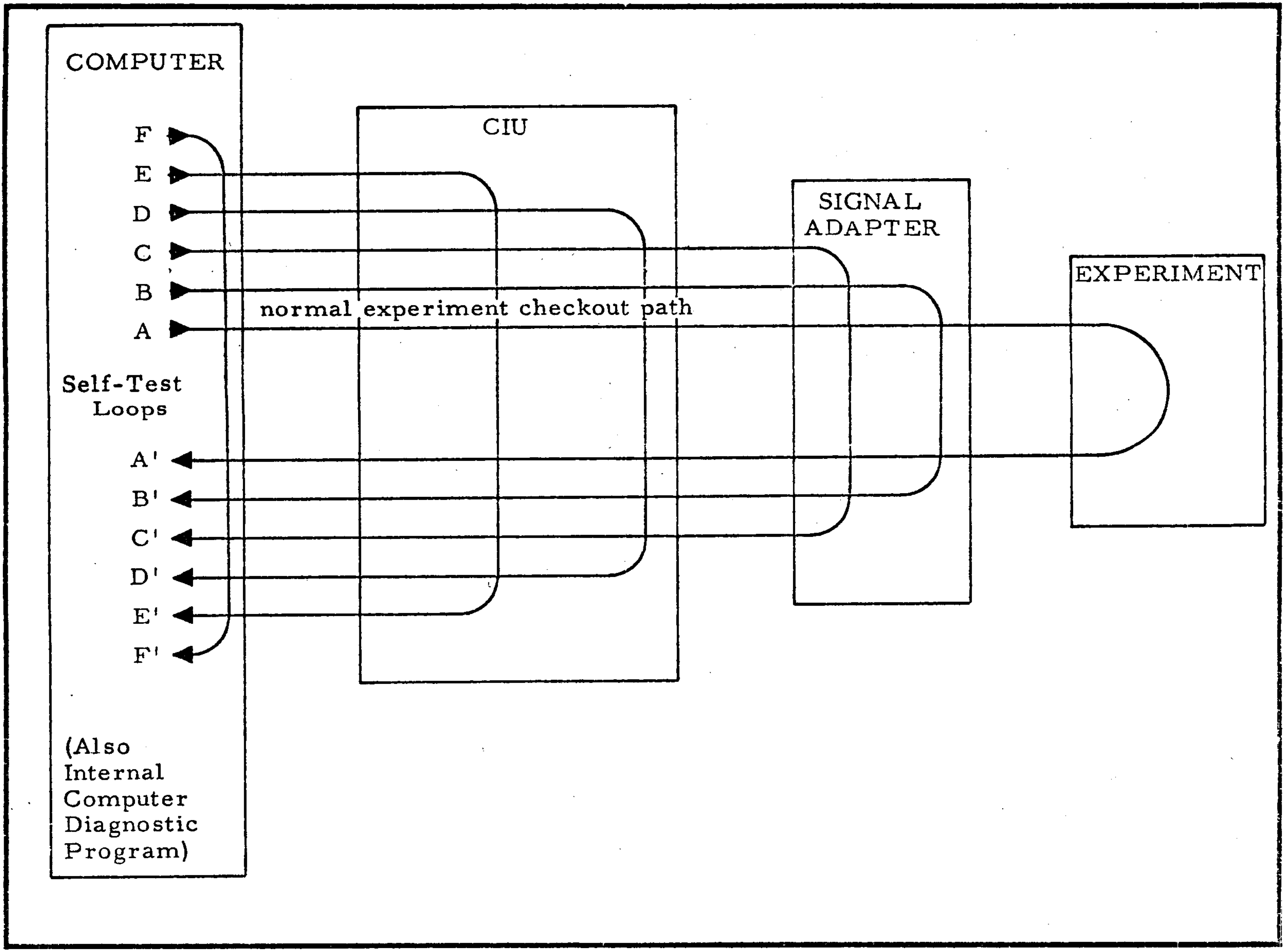

COMPUTER
F
E
D
C
B
A
Self-Test Loops
A'
B'
C'
D'
E'
F'
(Also Internal Computer Diagnostic Program)
CIU
SIGNAL ADAPTER
EXPERIMENT
normal experiment checkout path

EXHIBIT 13 - SELF-TEST SYSTEM

for OCDMS. These paths are provided in the hardware in order to achieve fault isolation to at least the functional unit level (i.e., CIU, DCU, etc.).

When the self-test mode is entered, the computer first checks itself for proper operation. A diagnostic program will have to be written for this purpose since manufacturer supplied diagnostics are rarely adaptable to an in-space fault-isolation scheme. The form this program takes and the manner in which it is initiated by the astronaut are complex subjects and beyond the intent of this analysis. It is sufficient for present purposes to assume that an adequate check on computer operations will be made prior to proceeding to test the rest of OCDMS.

After computer operation is verified, the loop FF' is tested. This is an internal (to the computer box) routing of outputs (normally going to the CIU) back into the input section. In effect, an I/O instruction is output to the input buffer channel. This tests most of the internal I/O logic and registers.

The next loop, EE', envisions transmitting I/O instructions to the CIU and then examining the CIU registers involved to see if they contain the proper data. Exhibit 2 shows some of the instructions available for this purpose.

Also internal to the CIU is loop DD'. Here the output of the Serial Output Register is fed back to the Serial Input Register as if it were another external device. When this loop has been checked, the CIU is deemed to be working with a high degree of confidence.

The first loop into each external device, CC', is used to check the I/O logic of that device. Here the contents of the Receiving Register are routed directly back via the Transmitting Register without initiating any operation within the external device.

The final self-test loop is BB'. In the case of the Signal Adapters, examples of this would be routing a D-A output directly to an A-D input or closing a discrete and sensing it with the discrete measurement capability. For the CDU in general, the astronaut is required to close the loop. For example, the computer could display a string of EL characters in a predetermined sequence and the astronaut could then observe and key back successful/unsuccessful indications.

The last loop is the normal checkout path and would be used only after successful operation had been achieved over the other paths.

Although these paths have been discussed as if each were singular, this is not the case. For example, there should be more than one self-test path from D-A to A-D or from discrete out to discrete in.

Slightly more isolation can be achieved if the second loop in each unit is external to the unit. That is, the connection is made via external cabling rather than internal. This approach includes more logic and also the connectors. However, it requires more external connections and wiring. More thought should be given to this method before employing it.

III. DIGITAL DATA TRANSMISSION

Because of the unknown and arbitrary distance, separately physical units of OCDMS (most notably the remotely located signal adapters), the digital data transmission between these units must be carefully analyzed. In this section a data transmission system between the CIU and external devices (i. e., signal adapters, CDU, etc.), of the OCDMS is described in general terms. The system is estimated to be capable of roles in the vicinity of 500 kc over cables of virtually unlimited length by efficiently taking into account transmission-line effects.

A. Isolation

One of the primary causes of problems in transmitting digital data from one equipment to another in a typical aircraft or spacecraft system is common-mode voltage differences, both transient and steady-state, between the equipments. Commonly, the boxes of a system are individually referenced (grounded) to the structure of the vehicle at a point near the box. Under such conditions, it is difficult to maintain small point-to-point voltage differences due to occasional large current transients within the structure between high-power equipments and the prime-power source, and electrical discontinuities within the structure. The problem is augmented where high-bit rates and minimal power are desired, since this implies transmission at relatively low-signal levels (i. e., levels comparable to the expected point-to-point potential differences within the structure).

Though there has been a growing awareness of the problem in recent years, it is not likely that a completely minimal solution will be forthcoming due to the highly statistical nature of the parameters identified. As a result, the only acceptable solution would be one immune to vitually any magnitude of common-mode voltage. The best way of achieving this with readily available elements at this point is by the use of an isolation transformer.

B. Coding

If an isolation transformer is used, the data must be coded in one of the A-C formats, the best of which is the so-called Manchester-Ferranti, or phase-shift technique. With this method the ratio of the fundamental frequency components of the modulation is only two, and the circuitry requirements for coding and decoding are quite minimal.

Exhibit 14 shows an arbitrarily chosen set of serial data as it would be coded into a transmission signal (f_t). For this example, a system has been chosen with a symmetrical clock for illustration. The essential characteristic of the transmission signal is that a transmitted zero is one polarity in the first half of a bit time and the opposite polarity in the second half, while a one reverses this phasing. Note particularly that under no conditions does f_t remain in any one state longer than one-clock cycle, nor less than one-half-clock cycle. It is this characteristic which makes this form of coding suited to transmission through an A-C-coupled device (e.g., a transformer). As shown in the example, the coding can be accomplished ideally by taking the modulo-2 sum (exclusive or) of the clock and the data--a simple combinational operation requiring three two-input gates--assuming availability of the negatives of the clock and the data. However, for one or more of the following reasons, it is desirable to perform the coding as a sequential operation employing a flip-flop.

1. Few systems utilize an exactly symmetricai clock. Symmetry is desirable in order to limit the fundamental frequencies in the transmission signal to within one octave of one another.

2. The simple combination method shown is subject to generating logic "slivers" at the points where the data signal changes state, due to slight differences in propagation time of the gates used. While such slivers may do no harm in an ideal system, they should be avoided on EMI grounds.

3. Decoding of the transmitted signal at the receiving end is conveniently done if a one-quarter-cycle time delay can be introduced.

It is desirable to perform the coding using elements and conventions which are standard to the rest of the system; hence, the

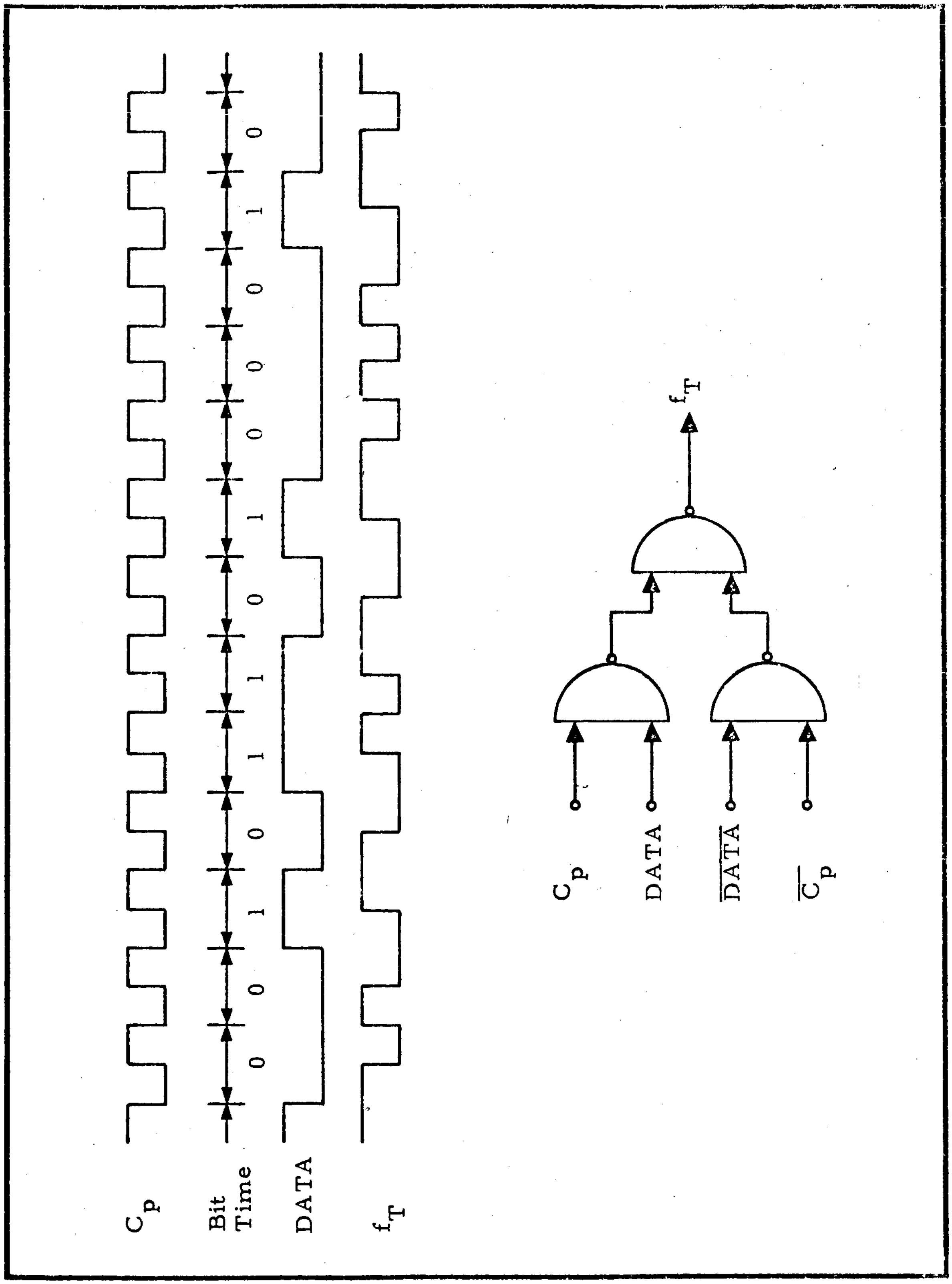


EXHIBIT 14 - MANCHESTER-FERRANTI CODING

details of the coding mechanization will depend on such system considerations as the type of clock available and the type of flip-flop used. Exhibit 15 shows one possible sequential method, wherein a pair of three-input NAND gates have been hooked as a binary toggle. The delay signals T and T* are generated from the negative transitions of the clock by one-quarter-clock-cycle one-shots. One-shots of this type are available in integrated circuit form. The delay period is set by the addition of an external capacitor (about one picofarad per nanosecond). The clock, C_p , is the normal system clock, and is used to operate the buffer storage register, as well as the rest of the system. The data, D , to be coded and transmitted, are shifted through the coding toggle. Exhibit 15 also presents some of the circuit waveforms for an arbitrarily chosen data stream. Although not specifically recommended here, there is no reason why the output gate of the toggle cannot double as the transmission line driver, described later. The transmitted data are delayed by the period of the one-shots, chosen here as one-quarter-clock cycle.

C. Line Driver

Historically, because of line capacitance problems which have been one of the chief limitations on speed in digital systems, a line driver has been thought of as a circuit capable of driving capacitance by brute force. Seldom has the concept of a line as a transmission line, with all the implicit consequences, been invoked. This has resulted in a waste of power and less than optimum performance. In this section, a line driver is discussed which is efficient, simple, composed of readily available components, and capable of optimum performance.

The primary undesirable characteristic of transmission lines is that once a line attains a length which is long, or at least comparable to a fraction of a wavelength of the higher frequency components of a digital signal, it exhibits reflections unless it is somewhere terminated in a controlled impedance equal to the characteristics impedance of the line. With high-speed transmission over long lines, these echoes

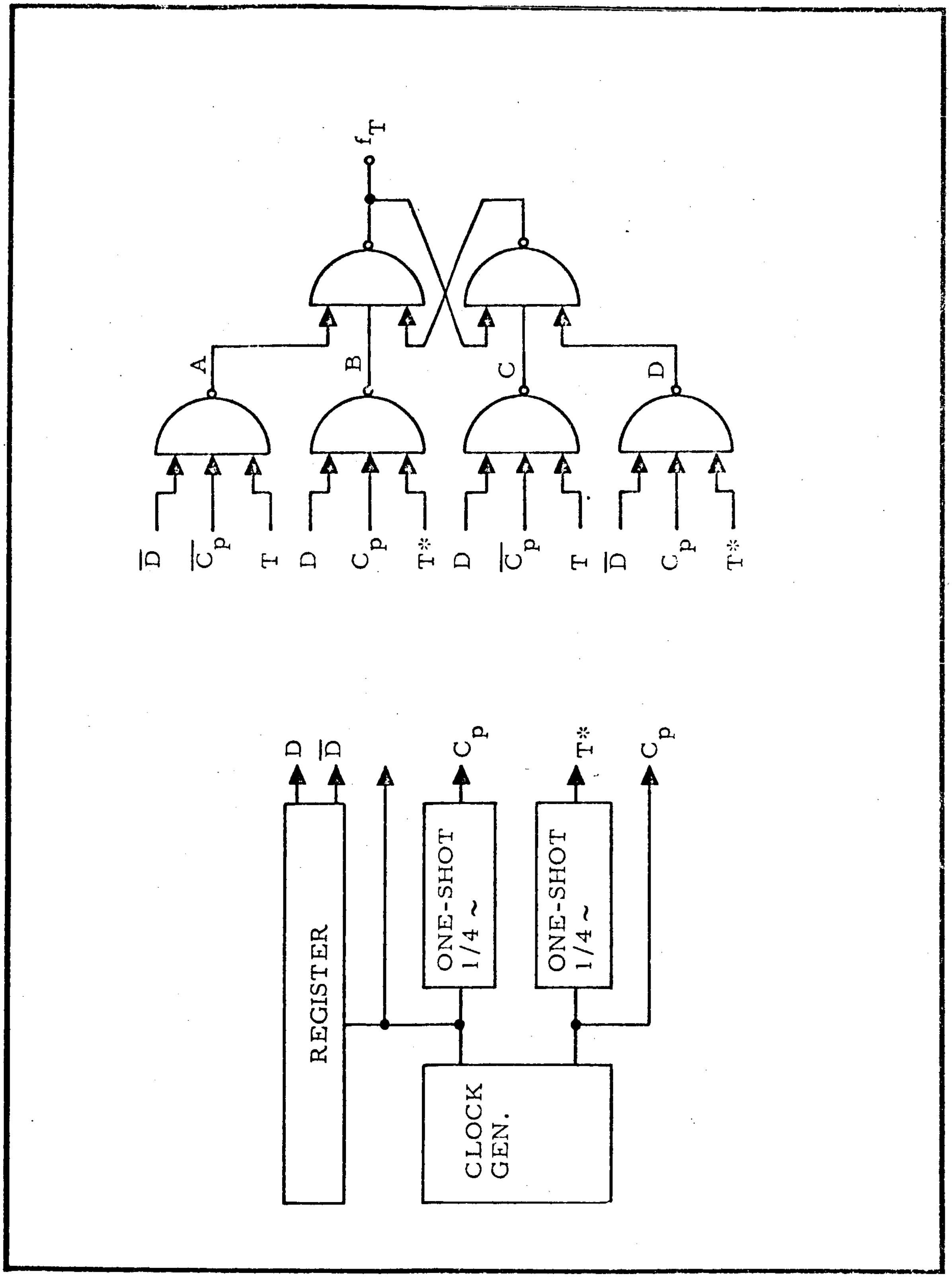


EXHIBIT 15 - A SEQUENTIAL METHOD

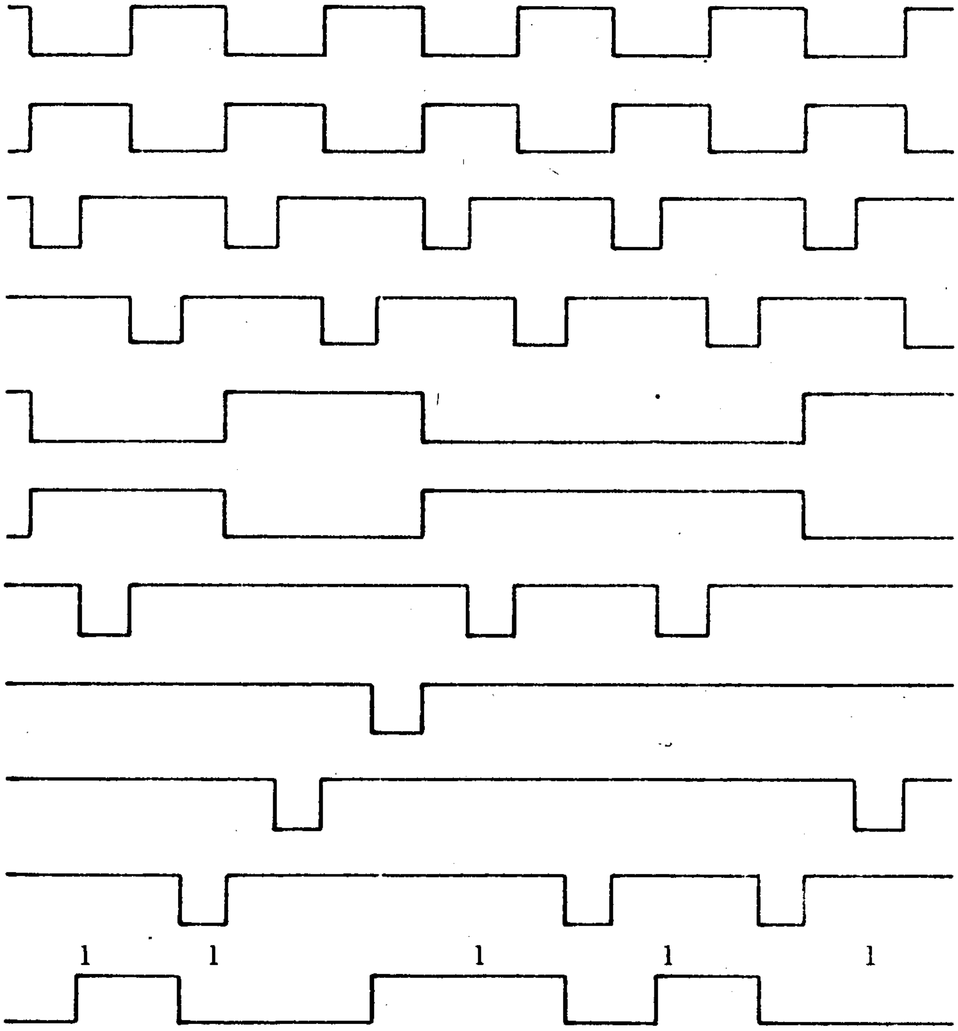


EXHIBIT 15 (Continued)

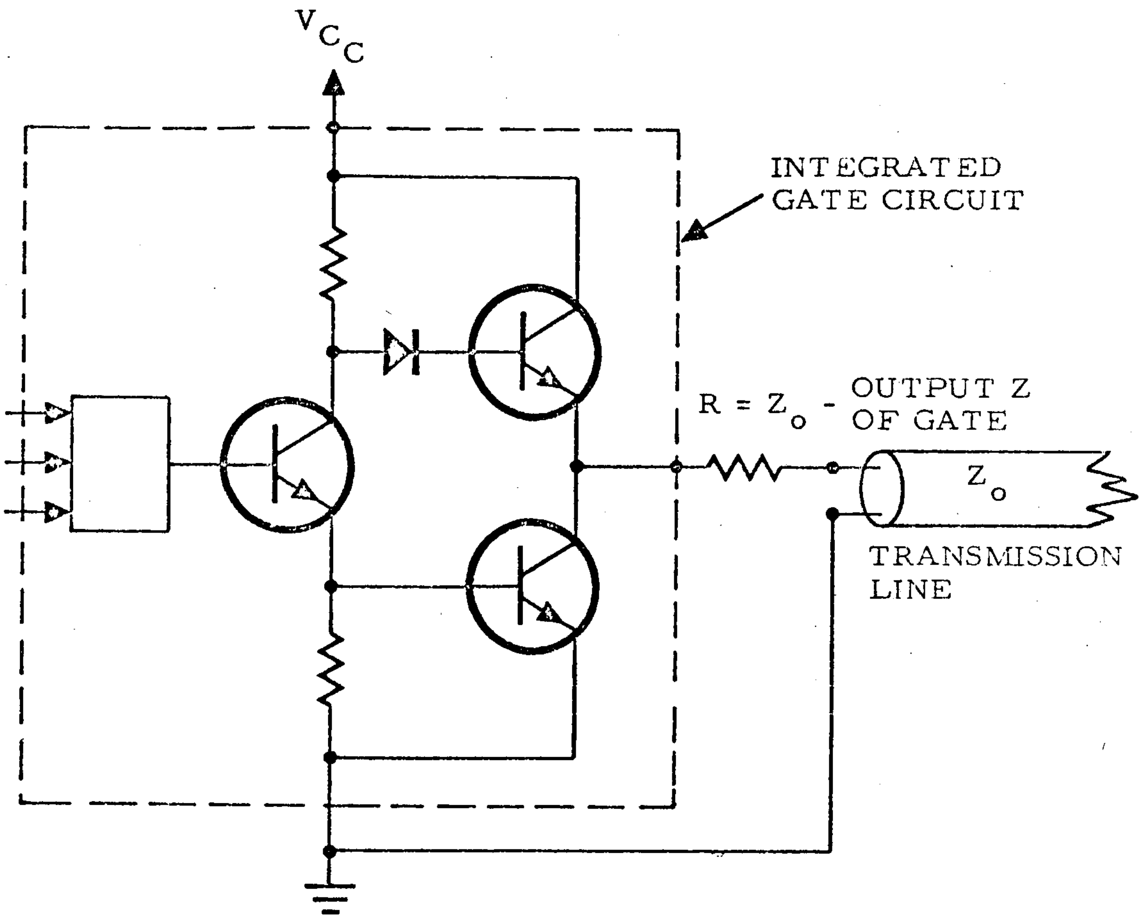
amount to system noise, which candegrade operational reliability. The proposed system provides termination at the transmission end of the line.

Exhibit 16 shows a line driver which is adapted from a standard type of integrated circuit gate by the addition of a resistor in series with the output terminal. There are many detailed variations on the integrated circuit shown available from most major semiconductor manufacturers. Basically, it is that associated with TTL. The essential feature is that it must have active pullup, rather than a merely passive collector load resistor. Active pullup permits the circuit to display a low-output impedance (10-30 ohms) in both directions of logic-level swing. A discrete series resistor is added between the integrated circuit and the line such that the net output impedance of the total circuit just matches that of the line. Though this resistor ideally could be incorporated within the integrated circuit, it is difficult in practice, since the ±25-percent tolerance which appears to be the present state of the art for diffused resistors is not adequate to ensure a satisfactory match to the line.

D. Transmission Line

Any wire or pair of wires which is comparable to or longer than the frequencies within it, is a transmission line. It has a characteristic impedance, Zo, which is determined by its cross-sectional geometry. The Zo of a single wire is difficult to determine, unless it is considered to exist in an otherwise void universe. In order to have a controlled Zo, it is therefore necessary to run transmission pairs. Twisting helps to give magnetic isolation by providing a large number of small-area interfering magnetic loops. In addition, twisting tends to ensure a reasonably uniform spatial relationship between the conductors, and, hence, a good control on Zo.

A line suitable to the present purpose may be constructed of practically any common wire, provided that the gauge and insulation thickness are uniform. To prevent interference and radiation, the pairs should be twisted to about one to three cycles per inch. Experience indicates that shielding of pairs as a group is necessary to prevent EMI to the satisfaction of virtually all government specifications. All



Note: The gate must be of the type with "active pull-up."

EXHIBIT 16 - LINE DRIVER

value has been chosen merely to correspond to common modern logic levels.) The generator is connected to the line through a series impedance equal to that of the line, say 100 ohms. The line here may be considered to be any length and is terminated by a receiver which has a differential input impedance which is considerably larger, perhaps by several kilohms, than that of the line.

Exhibit 17 also shows voltage waveforms at various points along the line. V_S is considered to execute a +4-volt transition at time $t = 0$. Point A, which for the period that the line is active can be considered to be loaded by Zo, undergoes a step one half the amplitude of V_S, since it represents the output mode of a 2:1 voltage attenuator. Assume the line to be cut to a length such that its one-way signal transit time is T. (Typical lines have propagation velocities about 60 percent that of light in free space, or about 6×10^8 feet per second.) This two-volt step impinges upon the impedance discontinuity represented by the high-input impedance of the receiver and a reflection occurs which is approximately the same amplitude and polarity as the incident step. By the principle of superposition, the reflection sums with the voltage to which the line has already been raised. Since incidence and reflection are sumultaneous at point D, the receiver passes point C slightly afterward, and passes point B at time 1.5 T. When the reflection reaches the driver, no further reflection occurs, since the matched output impedance presented by the driver guarantees no impedance discontinutiy.

For an opposite polarity transition (+4 to zero volts) the series of events would be entirely similar to that described above. One consequence of transmission-line theory is that the eventual steady-state condition of the input terminals must be exactly the same as in the case where the line is removed and the termination is connected directly to the generator, the only difference being the events which occur in the meantime. This is the case in the description above and would be the case for any value of termination including short-circuit. The significant feature of the proposed system is that a virtually full-amplitude signal is received, despite the impedance of the line.

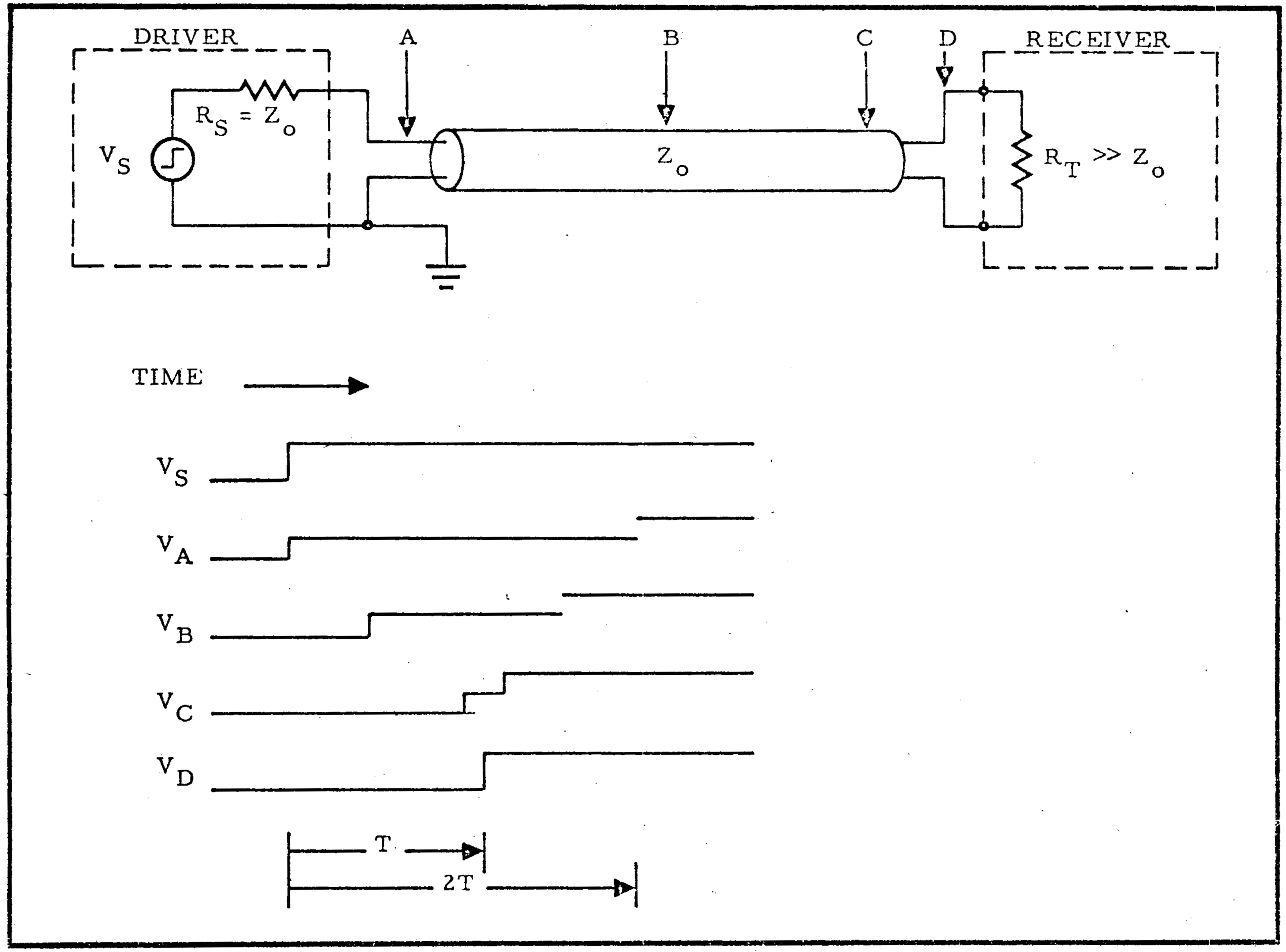


EXHIBIT 17 - TRANSMISSION CHANNEL MODEL

wires which are not members of the class herein described can be excluded from this shield enclosure.

The general expression for the Zo of any transmission line is $Zo = L/C$ where L and C are the inductance and capacitance per unit length of the line, respectively. For a two-wire pair Zo is $Zo = 120 \text{ Cosh}^{-1} D/d$ where D is the center-to-center spacing of the conductors and d is the conductor diameter. Practically all lines made with common types of wires usually end up having Zo very near the vicinity of 100 ohms, and variations in gauge and insulation thickness have relatively small effect.

Transmission lines have the property that a disturbance propagated toward an impedance discontinuity is reflected from the discontinuity according to the relation:

$$\frac{Vr}{Vi} = \frac{\frac{R_T}{Zo} - 1}{\frac{R_T}{Zo} + 1}$$

where Vr/Vi is the amplitude ratio of the reflected to the incident waves, and R_T/Zo is the impedance mismatch ratio. Note that for $R_T = Zo$ the reflection is zero (i. e., the line is terminated in a matched impedance), and the incident wave is completely absorbed. For R_T very large compared with Zo, the amplitude of the reflection is equal to and of the same polarity as the incident wave. The opposite case is where $R_T = 0$ (i. e., a shorted termination), when the reflection is equal in amplitude but opposite in polarity to the incident wave.

E. Transmission System, General

The proposed transmission system takes account of the actual properties of its transmission line. A model of the transmission channel is shown in Exhibit 17. A voltage generator of low impedance is capable of executing sharp transitions from zero to +4 volts. (This

If in the normal course of transmitting digital data the driver should execute a new transition, or even a whole series of transitions before the reflection of the first transition returns to the driver, as would occur with a very long line, operation at the receiver would not be affected, even though the waveform at the line input might look quite strange. This is because waves which propagate through one another on a line have no effect on one another except to instantaneously sum. Hence, this system can be used without signal distortion with any length line (up to the point where the sheer copper resistance of the line becomes significant) or any data frequency at which the circuits themselves are adequate. The only effect is the time delay due to the propagation velocity of the line.

Fanout to two or more different lines from a single driver is possible under three combined provisions.

1. That a separate Zo matching Rx is provided from the generator to each line.
2. That the output impedance of the generator is negligible.
3. That the driver circuit is capable of supporting the additional transient load.

The last two conditions are not usually fulfilled by commonly available integrated circuit gates; hence, this expedient is not recommended. the other hand, fanout from a single line into several receivers would be permissible as long as the combined impedance is still large compared with Zo .

F. Receiver

One of the fundamental considerations in any digital system is the amount of available noise margin provided by the system's gate circuitry. The nature of digital operation, in contrast to analog, is such that a single-transient-erroneous change of logic state can lead to a completely wrong result.

An example of the effect of the operating parameters of a given gate on the noise margin is shown in Exhibit 18, which plots the input characteristics of a hypothetical integrated circuit gate as a function

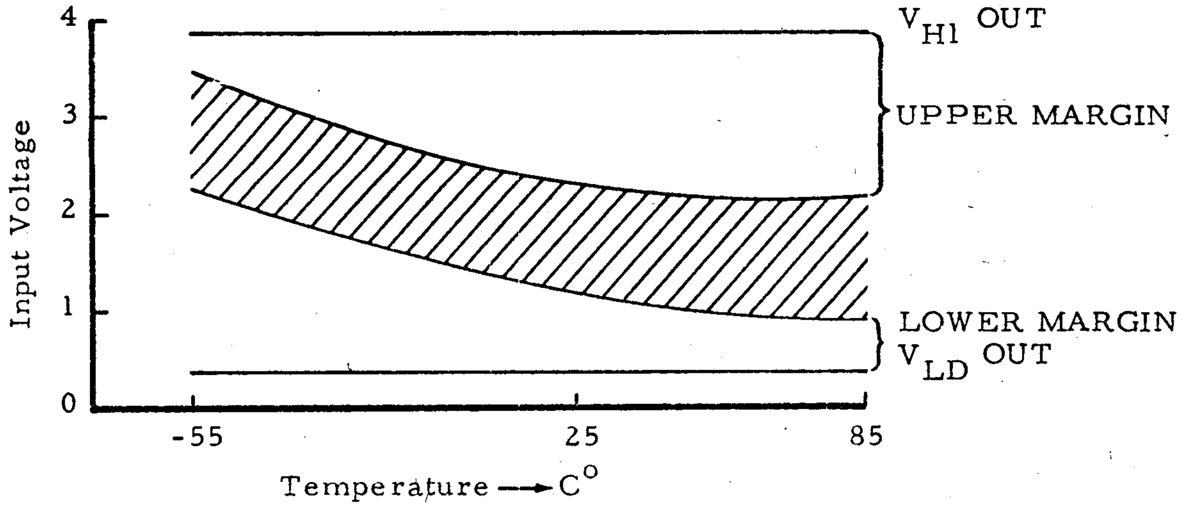


EXHIBIT 18 - TYPICAL INPUT CHARACTERISTICS, HYPOTHETICAL INTEGRATED CIRCUIT GATE

of temperature. Any gate has a certain range of input voltage for which the output state is indeterminate. The range of indeterminateness is a combined function of variations in the circuit gain, output loading, and variations in the components from unit to unit. In addition, one of the chief variations is due to temperature. Also plotted are the maximum expected zero output voltage and the minimum one output voltage from a similar gate, or any other circuit which might be expected to drive the gate in question. The regions between these levels and the indeterminate zero are considered the noise margins; a voltage perturbation at a given temperature of this amount may cause a logic error. As can be seen, it is well to design the circuit so that the indeterminate level is as near to the center of the nominal logic levels as possible for the sake of noise immunity.

Noise margin has been a subject of concern in the past few years with stress being placed on speed and minimization of power. Thus, practically all available integrated logic circuitry embodies some compromise among these elements. For this reason it is possible that a standard logic circuit suitable for use within a unit would not be suitable as a line receiver between units, where a considerably greater degree of differential noise voltage could be expected.

In addition to the consideration of noise immunity is the question of input impedance. All logic circuits achieve their relatively sharp switching actions by the use of P-N junctions in the input biasing circuitry. This implies radical changes in input impedance as a function of input voltage. In contrast, the proposed system requires that the line receiver possess a high-input impedance.

The considerations discussed indicate that a suitable circuit should have a high-input impedance, and a relatively narrow and controlled range of input-voltage sensitivity, preferably one which is constant with temperature. The general class of circuit which satisfies these characteristics is the differential amplifier. In the field of available integrated circuitry, the differential amplifier has been adapted to the form of high-speed voltage comparators and sense amplifiers for use in core memories. An example is the Fairchild UA710.

This circuit has an input switching sensitivity of several millivolts and is virtually independent of temperature. It switches at nominally zero volts input differential and has a considerable degree of common-mode noise rejection. Its speed is comparable to half that of most DTL or TTL gates.

The proposed line receiver would have an input transformer to achieve the unit-to-unit isolation previously discussed. The transformer would have a primary-to-secondary voltage stepdown of about 40:1, with a center-tapped secondary (the center tap being tied to local ground). The effect is to transform a zero to +4-volt logic swing to a -100 to +100 millivolts differential input to the differential amplifier. The transformation ratio is also such that a very high impedance is presented to the line. In addition, the transformer must have a high primary impedance at the data frequency, together with a satisfactory high-frequency response. It is roughly estimated that a suitable pulse transformer could be packaged within a cube one-half inch on a side.

G. Ferranti Decoding

Decoding of Ferranti format information into conventional binary serial data is relatively simple. Reference to Exhibit 15 shows that the conventional clock, Cp , undergoes a negative transition coincident with a point three-quarters through a transmission data bit time because the coding method causes a one-quarter-cycle delay. Hence, if the Cp clock is transmitted over a separate channel identical with the data channel, including line driver, transmission line, transformer, and receiver, it may be used to clock the data into a buffer register at the receiving end. If a flip-flop which triggers at a negative clock transition is used, this results in conversion into normal serial binary format. This flip-flop would be the input cell of the buffer register.

H. System Timing

In transmitting data from one unit to another, synchronism must be maintained between the data and the clock, which is transmitted for the purpose of synchronously decoding the data at the terminus. But the situation here is little different than that in which point-to-point transmission occurs in any coding format. In systems where the two units have independent clocks, one has the normal logic design problem of traffic buffering. This requirement to maintain terminal-end clock/data synchronism is the primary consideration in limiting the transmission frequency, and, hence, the data rate. Exhibit 19 outlines a model of a hypothetical system. Consider for the moment only the channels which flow from the central unit to the remote unit. If the central clock is taken as the time reference, by the time it is used to decode the transmitted data, it suffers propagation delay in the line driver, the line, the transformer at the input of the line receiver, the line receiver itself, and possibly in a stage of buffer amplification. The path of the data subjects it to the propagation delays of the one-shot, the coder/line driver, the line, and the line receiver and transformer Since most of the one-shot delay of one-quarter cycle is intentional, ±10 percent or ±25 nanoseconds, whichever is greater, can be assigned as a conservative estimate of its contribution to synchronization error. The line itself can be completely discounted--assuming equal lengths of identical line for each channel--and two lines can vary only a few percent in their propagation characteristics. A 50-foot line has a propagation time of about 100 nanoseconds, with a possible variation about the actual figure of perhaps ±5 nanoseconds. Modern logic circuitry has delay times in the 50 nanoseconds per stage region with a variation conservatively assigned at ±50 percent. Fifty nanoseconds ±20 for the transformer, and 100 nanoseconds ±50 for the receiver should be assigned.

Assuming a one megacycle system, this places the one-shot period at a nominal 250 nanoseconds. Taking the assigned values, it is found that each of the channels suffers a propagation delay of about 300 nanoseconds (discounting the line), with a variation in the vicinity of ±150 nanoseconds. For the system postulated, synchronism must be maintained within ±250 nanoseconds, (±one-quarter cycle) and, therefore, there is a margin of ±100 nanoseconds. In a real system the judgment as to whether this is adequate would depend on weighing such factors as the system purpose, the confidence with which the delay deviations for the actual elements of the system were placed, and probable additional complexities which the analysis has not foreseen.

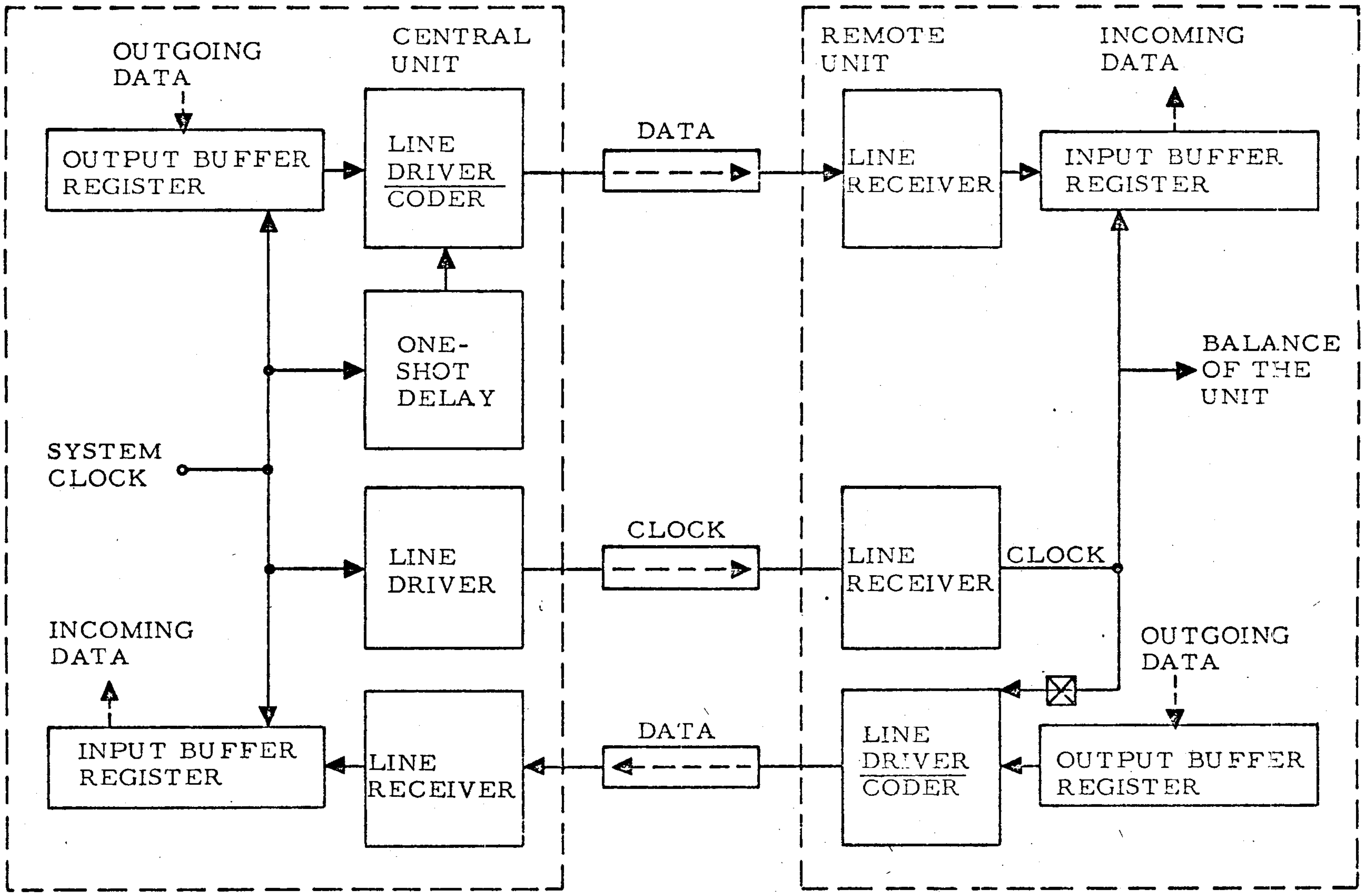


EXHIBIT 19 - BLOCK MODEL OF TRANSMISSION SYSTEM

Next, the additional problem of two-way transmission should be considered. Reference to Exhibit 19 shows that data arriving from the remote unit are nominally in zero phase with the local clock. This makes it tempting to place a one-shot at the place marked X in the remote unit in order to again effect a nominal one-quarter-cycle separation. However, it should be noted that these incoming data are subject to a two-way delay in arriving at the central unit, since they are transmitted synchronous with the clock as received at the remote unit. Since this total delay can never be negative, some unambiguous separation between the local clock and the data as received is always guaranteed so long as the total delay does not exceed one-half-clock cycle. Exhibit 20, item B, which lists the delays, shows that if the previously assumed figures are realistic, up to one microsecond delay could be experienced. Therefore, in order to avoid the extra hardware of a clock channel from the remote to the central unit, the clock frequency must be in the vicinity of one microsecond per half cycle, or 500 kc.

EXHIBIT 20 - PROPAGATION DELAYS

A. Transmission Central to Remote:

	Data	Clock
One-shot	--- ± 25	
Coder/Line Drive	100 ± 50	50 ± 25 (1 gate)
Line	--- ± 05	--- ± 05
Transformer	50 ± 20	50 ± 20
Line Receiver	100 ± 50	100 ± 50
Buffering (2 gates)		100 ± 50
	250 ±150 ns	300 ±150 ns

B. Transmission Remote to Central: (assumed 50-foot line)

Clock	
Line Driver	50 ± 25
Line	100 ± 05
Transformer	50 ± 20
Receiver	100 ± 50
Buffering (?)	100 ± 50

Data	
Line Driver/Coder	100 ± 50
Line	100 ± 05
Transformer	50 ± 20
Receiver	100 ± 50
	750 ±275