

FINAL REPORT
FOR
ACTUATOR DIGITAL INTERFACE UNIT (AIU)

N74-29527

(NASA-CR-120328) ACTUATOR DIGITAL
INTERFACE UNIT (AIU) Final Report
(Electronic Communications, Inc.)
HC \$25.50

288
448 P
CSCI 09B

Unclas
G3/08 15908

Reproduced from
best available copy.

ELECTRONIC COMMUNICATIONS, INC.
A SUBSIDIARY OF NCR



SYSTEM DEVELOPMENT
COMMUNICATIONS
DATA LINK
DATA PROCESSING
ANTENNAS
MULTIPLEXERS

ST. PETERSBURG, FLORIDA

ACTUATOR DIGITAL INTERFACE UNIT (AIU)

CONTRACT NUMBER NAS826755

Prepared for:

**National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama 35812**

Prepared by:

**E. C. I.
1501-72nd Street No.
St. Petersburg, Fla. 33733**

**FINAL REPORT
October 1973**

TABLE OF CONTENTS

	Page
1.0 INTRODUCTION	1
2.0 SYSTEMS FUNCTIONAL DESCRIPTIONS	2
2.1 Alternate "A" Functional Description	2
2.1.1 Mathematical Model Functional Description .	5
2.1.1.1 Dedicated Processor	5
2.1.1.1.1 General Description	5
2.1.1.1.2 Main Memory	7
2.1.1.1.3 Main Bus	8
2.1.1.1.4 Control Memory	8
2.1.1.1.5 Special Registers	9
2.1.1.1.6 Extend Register	9
2.1.1.1.7 Interrupt Register	12
2.1.1.1.8 Register File	12
2.1.1.1.9 Processor Interrupts	13
2.1.1.2 Microprogramming Language	14
2.1.1.3 Mathematical Model Solution	14
2.1.1.3.1 General Description	15
2.1.1.3.2 State Variable Equations	16
2.1.1.3.3 Numerical Solution	18
2.1.1.3.4 Scaling	19
2.1.1.3.5 Sampling Rate	19
2.1.1.3.6 Computer Program	19
2.1.1.4 Interrupt Processing	19

TABLE OF CONTENTS (Continued)

	Page
2.1.1.5 Transfer Function Processing	22
2.1.1.6 Comparison Routine	22
2.1.1.7 Hydraulic Management	23
2.1.1.7.1 General Description	23
2.1.1.7.2 Detail Description	23
2.1.1.8 Diagnostic Processing.	28
2.1.1.9 Maintenance Panel	28
2.1.1.9.1 Purpose of Maintenance Panel	28
2.1.1.9.2 Detailed Explanation of Maintenance Panel Switches.	29
2.1.1.9.3 Maintenance Panel Operation	32
2.1.1.10 Paper Tape Generation	35
2.1.2 Simplex Functions	36
2.1.3 Redundant Functions	38
2.2 Alternate "B" Functional Description	38
2.3 Test Set Functional Description	41
2.3.1 Test Set System Operation.	46
2.3.1.1 Timing Logic.	46
2.3.1.2 Supervisory Word Generator	46
2.3.1.3 Data Word Generator	49
2.3.1.4 Clock Generator and A/D Converter.	49
2.3.1.5 Display Panel Status Monitor	49
2.3.1.6 Display Panel Switch Logic	53
2.3.1.7 Actuator Analog Math Model	55
2.3.2 Operational Considerations	55
3.0 SYSTEMS ELECTRICAL DESIGN	59
3.1 Alternate "A" Circuit Modules	61
3.1.1 Line Receiver; Clock & Data Bit Counter/ Module No. 148; Loc B1	61

TABLE OF CONTENTS (Continued)

		Page
3.1.2	One's Supervisory Serial & Storage Registers/Module No. 6; Loc B2	63
3.1.3	Supervisory Decoder & Parity Checker/ Module No. 11; Loc B3	63
3.1.4	Data Serial & Storage Registers/ Module No. 105; Location A1	66
3.1.5	Servo Amplifier; A/D and D/A Converters/ Module No. 29; Locations A3, A6, and A9 .	66
3.1.6	Eight-Bit Comparator ± 2 Bits/Module No. 58; Locations A2, A5 & A8	69
3.1.7	Power Interrupt Logic/Module No. 60; Locations A4, A7 & A10	71
3.1.8	Computer Input Logic/Module No. 74; Location C3	73
3.1.9	Computer Output Logic/Module No. 75; Location C2	73
3.1.10	Computer Clock Generator/Module No. 69; Location B11	75
3.1.11	Scratch Pad Memory/Module No. 81; Location C4	78
3.1.12	Random Access Control Storage/Module No. 82; Locations C7 & C8 Random Access Control Storage/Module No. 83; Location C9	80
3.1.13	Memory Access Multiplexer/Module No. 73; Location C10	80

TABLE OF CONTENTS (Continued)

		Page
3.1.14	Parity Generator & Check Logic/Module No. 72; Location C6	84
3.1.15	Tape Reader Interface Clock/Module No. 70; Location C12	84
3.1.16	Tape Reader Interface; Data Acquisition and Verification Logic/Module No. 71; Location C11	84
3.1.17	Hydraulic Switch Drivers/Module No. 62; Location C1	89
3.1.18	Dual Tracking A/D Converter/Module No. 20; Location B10	91
3.1.19	Status Monitor Timing & Control/Module No. 61; Location B4	91
3.1.20	Status Monitor Data Multiplexer/Module No. 17; Location B6 & B8	94
3.1.21	Status Monitor Parity Generator; Shift Regis- ter; Polar RZ Modulator/ Module No. 114; Location B7	94
3.1.22	Alternate "A" Power Supply/Module No. 64/ Location B13	94
3.2	Alternate "B" Circuit Modules	99
3.2.1	Line Receiver/Module No. 48; Locations A3 & B3	99
3.2.2	Serial Voter/Module No. 4; Locations A5, A6, A8, B5, B6, C5 & C6	99
3.2.3	Systems Clock Generator/Module No. 19; Location B7	102

TABLE OF CONTENTS (Continued)

		Page
3.2.4	Supervisory Serial & Storage Registers/ Modules 6 & 7; Locations C1, C2, D1, D2, C7, C8, C11 & C12	102
3.2.5	Supervisory Parity Checker/Module No. 2; Locations C3, D3, C9, & C13	105
3.2.6	Alternate "B" Supervisory Decoder/ Module 12; Locations C4, D4, C10, D14 . . .	105
3.2.7	Single-Bit Majority Voter Summer/Module No. 3; Locations D5, D6, D7, D8, D9, D10 D11, D12, D13, D14, A11, A12, A13, A14, B11, B12, B13 & B14	105
3.2.8	Data Word Serial & Storage Registers/ Module No. 5; Locations A9, A10, B9, & B10	109
3.2.9	Data Bit Counters/Module No. 44; Location A7	109
3.2.10	Servo Amplifier & A/D Converter/Module No. 29; Locations E2, E4 & E6.	112
3.2.11	Eight-Bit Comparator ± 2 Bits/Module No. 58; Locations #1, E3 & E5	112
3.2.12	Power Interrupt/Module No. 46; Locations H2, H3 & H4	112
3.2.13	A/D Clocks/Module No. 31A; Location F8 . . .	112
3.2.14	Test Sequencer System	112
3.2.14.1	Test Sequencer Control Memory/Module No. 55; Location F1	115
3.2.14.2	ISVC Test Subroutine Logic/Module No. 51; Location F2	118

TABLE OF CONTENTS (Continued)

		Page
3.2.14.3	ISVC Subroutine Test Encoder/Module No. 52; Location F6	120
3.2.14.4	IDVC Test Subroutine Logic/Module No. 53; Location F3	122
3.2.14.5	Test Sequencer 8-Bit Comparator/Module No. 56; Location F4	124
3.2.14.6	Servo Linearity & IDVC Encoder Multiplexer	126
3.2.14.7	Status Monitor Subroutine Logic/Module No. 54; Location F7	128
3.2.15	Status Monitor Timing & Control/Module No. 45; Locations F9, F10, F11 & F12 . . .	128
3.2.16	Status Monitor Multiplexer/Module No. 15; Locations E9, E10, E11, E12, G9, G10, G11 & G12	131
3.2.17	Status Monitor Parity Generator & Shift Register/Module No. 14; Locations E13 & G13	131
3.2.18	Polar RZ Modulator/Module No. 16; Location F13	131
3.2.19	Alternate "B" Power Supplies A, B & C/ Module No. 49; Locations H7, H9 & H11. . .	135
3.2.20	Alternate "B" Power Supply D/Module No. 50; Location H13	135
4.0	CONCLUSIONS	138
Appendix A	General Derivation of Second Order Runge-Kutta	A-1
Appendix B	Flow Diagram of Computer Program	B-1

TABLE OF CONTENTS (Continued)

		Page
Appendix C	Computer Program Listing	C-1
Appendix D	Scaling Example	D-1
Appendix E	Listing of 1130 Paper Tape Program	E-1
Appendix F	μ - Mite Description of Assembly Language for the Programmable Controller	F-1
Appendix G	Math Model Processor & Maintenance Panel Electronics.	G-1

LIST OF TABLES

		Page
Table 2-1	List of Variables	20
Table 2-2	List of Constants	21
Table 2-3	Actuator Interface Unit(s) Supvr. Commands.	58
Table 3-1	Perforated Tape Format	87
Table 3-2	Alternate "B" Self Test Control Memory Program	117

LIST OF ILLUSTRATIONS

Figure		Page
1	Actuator Interface Unit; Alternate "A"	3
2	Hydraulic Valve Switch Configuration Actuator Unit Alternate "A"	6
2-1	Actuator Interface Alternate "A" Mathematical Model Processor Block Diagram	10
2-2	Arithmetic Unit	11
2-3	Hydraulic Actuator Transfer Function	15
2-4	Hydraulic Actuator State Variable Model	17
2-5	Hydraulic Redundancy Management Program Flow Diagram	24
2-6	Paper Tape Format	37
3	Actuator Interface Unit; Alternate "B"	39
4	Test Set Block Diagram	42
4-1	APS/DIU Test Set	43
5	Test Set Timing Logic	47
6	Supervisory Word Generator	48
7	Data Generator	50
8	Clock Generator & A/D Converter	51
9	Display Panel Status Monitor	52
10	Display Panel Switch Logic	54
11	Actuator Analog Math Model	56
12	Math Model	57
12-1	Circuit Module Coordinate System	60
13	AIU Alt "A" Breadboard Line Receivers.	62
14	AIU "A" Supervisory Registers.	64
15	AIU "A" Supervisory Decoder	65

LIST OF ILLUSTRATIONS (Continued)

Figure		Page
16	AIU "A" Data Register	67
17	Servo Amplifier	68
18	8-Bit ± 2 -Bit Comparator	70
19	Power Interrupt Module	72
20	Computer Input Logic	74
21	Computer Output Logic	76
22	Computer Clock Generator.	77
23	Scratch Pad Memory	79
24	Random Access Control Storage #1 & #2	81
25	Random Access Control Storage #3	82
26	Memory Access Multiplexer	83
27	Parity Generator/Check Logic	85
28	Tape Reader Interface Clock	86
29	Tape Reader Data Acquisition	88
30	Hydraulic Switch Drivers	90
31	Dual A/D Converter	92
32	Status Monitor Timing & Control	93
33	Status Monitor Multiplexer.	95
34	Status Monitor Parity Generator, Shift Reg., & Polar RZ Modulator	96
35	Alt "A" Power Supply	98
36	Alt "B" Line Receiver.	100
37	Alt "B" Serial Voter	101
38	Alt "B" System Clock Generator	103
39	Alt "B" Supervisory Serial & Storage Registers	104
40	Alt "B" Supervisory Parity Checker	106
41	Alt "B" 1's & 0's Decoder	107

LIST OF ILLUSTRATIONS (Continued)

Figure		Page
42	Alt "B" Single Bit Majority Voter-Summer	108
43	Alt "B" Data Word Serial & Storage Registers	110
44	Alt "B" Data Bit Counters	111
45	Alt "B" Power Interrupt - A, B, & C	113
46	Alt "B" A/D Clocks	114
47	Alt "B" Test Seq. Control Memory	116
48	Alt "B" ISVC Test Subroutine Logic.	119
49	Alt "B" ISVC Test Encoder.	121
50	Alt "B" IDVC Test Subroutine Logic	123
51	Alt "B" Test Seq. 8-Bit Comparator	125
52	Alt "B" Servo Linearity & IDVC Encoder Multiplexer	127
53	Alt "B" Status Monitor Subroutine Test	129
54	Alt "B" Status Monitor Timing & Control	130
55	Alt "B" Board #15 Multiplexer	132
56	Alt "B" Status Monitor Parity Gen.& Shift Register	133
57	Alt "B" Polar RZ Modulator	134
58	Alt "B" Power Supply A, B & C	136
59	Alt "B" Power Supply "D"	137

Appendices

B-1	Flow Diagram of Interrupt Routine	B-1
B-2-1	Math Model Solution Processing	B-2
B-3	Comparison Routine	B-5
B-4	Shift Right Routine With Rounding	B-6
B-5	Double precision Shift Right With Rounding	B-7
B-6	Multiply Subroutine	B-8

ACTUATOR INTERFACE DIGITAL INTERFACE UNIT; ALTERNATES "A" AND "B"

FINAL REPORT

1.0 INTRODUCTION

On April 21, 1971, ECI was contracted to design, fabricate, and test two alternate versions of an Actuator Interface Unit intended as a user subsystem of the Space Shuttle Data Bus control system. This unit would be employed as a controller for hydraulic-coupled air control surfaces and/or hydraulically controlled steerable engines.

Two alternate versions of the Actuator Interface Unit were conceived by NASA. Alternate "A" is a dual-failure immune configuration which feeds a "look-and-switch" dual-failure immune hydraulic system. Alternate "B" is a single-failure immune configuration which feeds a majority voting hydraulic system. Both systems communicate with the data bus through Data Terminals dedicated to each user subsystem. Both operational control data and configuration control information are processed in and out of the subsystem via the Data Terminal which yields the Actuator Interface subsystem, self-managing within its failure immunity capability.

The system designs presented here represents a significant milestone in the state-of-the-art of control systems hardware, in that the majority voting and dual-failure immune digital concepts are demonstrated through the implementation of Complimentary MOS circuitry. Here, the levels of complexity and design techniques are proven, and the systems are basically ready to be implemented into an LSI architecture.

2.0 SYSTEMS FUNCTIONAL DESCRIPTIONS

This section of the report deals with the two alternate systems operational characteristics and their functional design philosophies.

2.1 ALTERNATE "A" FUNCTIONAL DESCRIPTION

Figure 1 is a simplified block diagram of the Alternate "A" system. Two information input ports are shown as "data word" and "supervisory word" respectively, which is the command and control output from the bus data terminal.

The "supervisory word" input is a continuous stream of 20-bit words in a polar RZ format with the 20th bit missing to establish word synchronization. The normal format of this word is all zero's unless a specific command word is sent. Then, the applicable code is distributed from bit 9 through 16. Bit 16 carries parity. The supervisory word is also used to provide the basic clock input to the system. Here, each bit is sensed either as a one or a zero and ored to provide the clock.

The "data word" is also formatted within the 20 bit word structure, but it appears only when a word is to be entered. No signal exists on this input when there is no data to be transferred, which is different from the supervisory all-zero quiescent mode. The data word occupies bits 11 through 19, where bit 19 is the parity bit. Again, the bit format is polar RZ.

The Status Word output from the Status Monitor is also a 20-bit word in polar RZ format, which exists only upon request via the supervisory link. Here, a specific set of information is requested by a supervisory word, and the status word is released 40 usec later back to the data terminal.

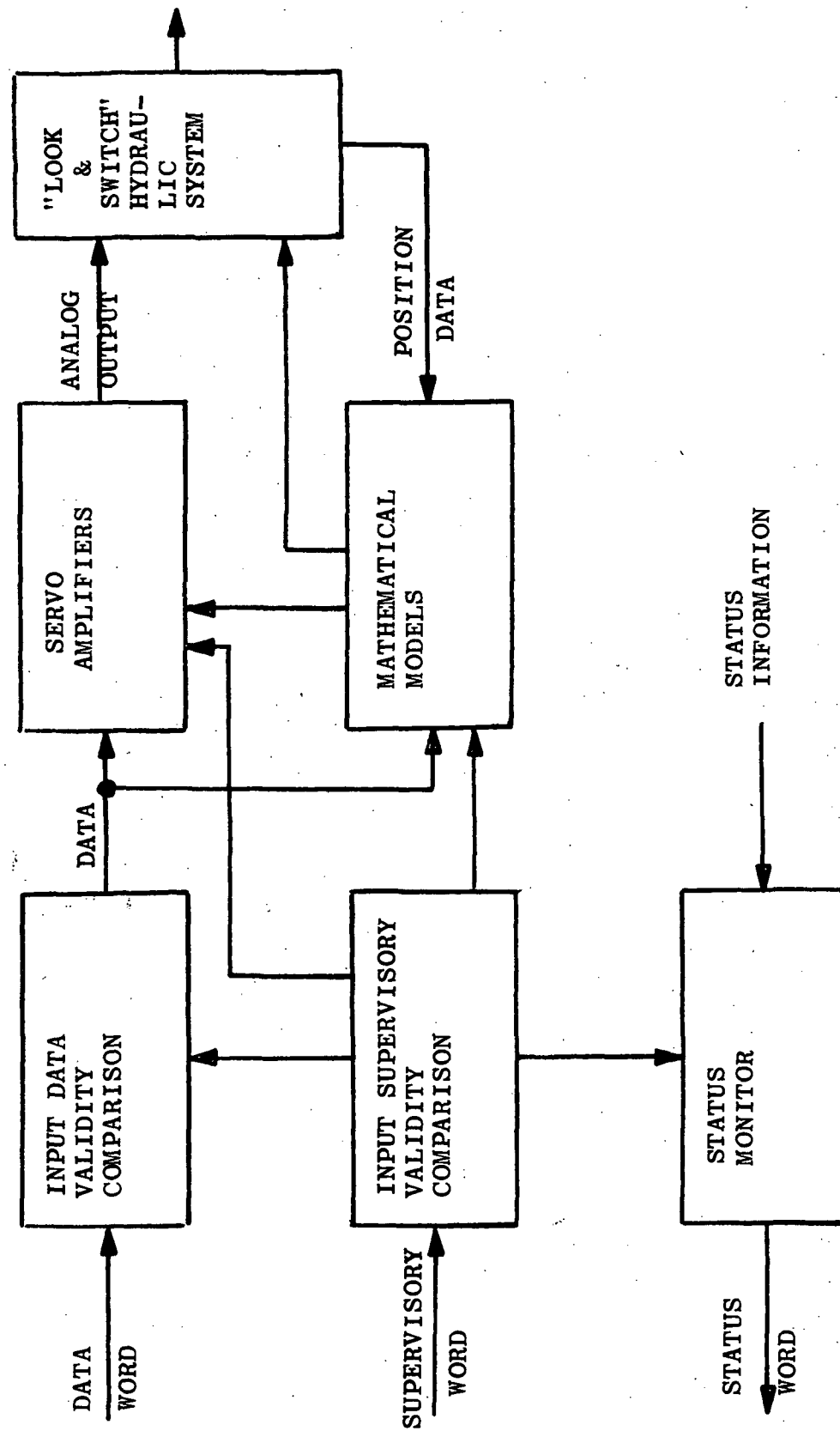


FIGURE 1 ACTUATOR INTERFACE UNIT; ALTERNATE "A"

Both the data and supervisory words are validated as they enter the system. Both serial and parallel voting is employed as well as parity tests. The output of both of these "validity comparison" functions is such that the data or command exists on a commoned three-wire bus-such that, at that point, the bit is immune to any two (or more) failures behind it. The data is fed to both the servo amplifiers and the mathematical models and this failure immunity insures that both receive and operate on identical data. The supervisory outputs are redundantly decoded in addition to the serial and parallel tests. Each supervisory command, then, exists with dual-failure immunity behind it also.

Both the Servo Amplifiers and the Mathematical Models process the same digital word. A digital-to-analog converter is provided for each of three servo amplifiers, and the outputs of the servos is fed to the hydraulic actuator system. Only one of the servo amplifiers is employed at a time, however, which is a function of the performance of the amplifier and it's associated hydraulic loop. This performance is monitored by the mathematical model section. Here, a digital computation of the hydraulic system's transfer function is continuously iterated, and the resultant answer is periodically compared with the actual performance of the actuator. Because of the self-tests incorporated within the math model, any dis-compare experienced between the calculated performance and the actual performance of the hydraulic system is accounted for as a failure or deterioration of the hydraulic system. A built-in "hydraulic redundancy management routine" then tests and controls the hydraulic loop to be installed to correct for the discompare error. The self-test mechanisms of the math model are sufficient diagnostics to deem the math model operational. Should a math model become inoperative,

a back-up model is installed to resume the identical functions of the first. Similarly, a third unit is available for further backup should the second fail.

The hydraulic system configuration is shown in Figure 2. The normal operational loop or prime loop is shown where servo amplifier "A" is "on-line". The first back-up utilizes servo amplifier "B" by shuttling hydraulic switch "A". The second back-up utilizes servo amplifier "C" by shuttling hydraulic switch "D". Hydraulic switch "B" is used to by-pass switch "D", should switch "D" inadvertently shuttle and servo amplifier "C" fail. This would place servo amplifier "A" (or "B") back on line where it was not intentionally removed because of failure.

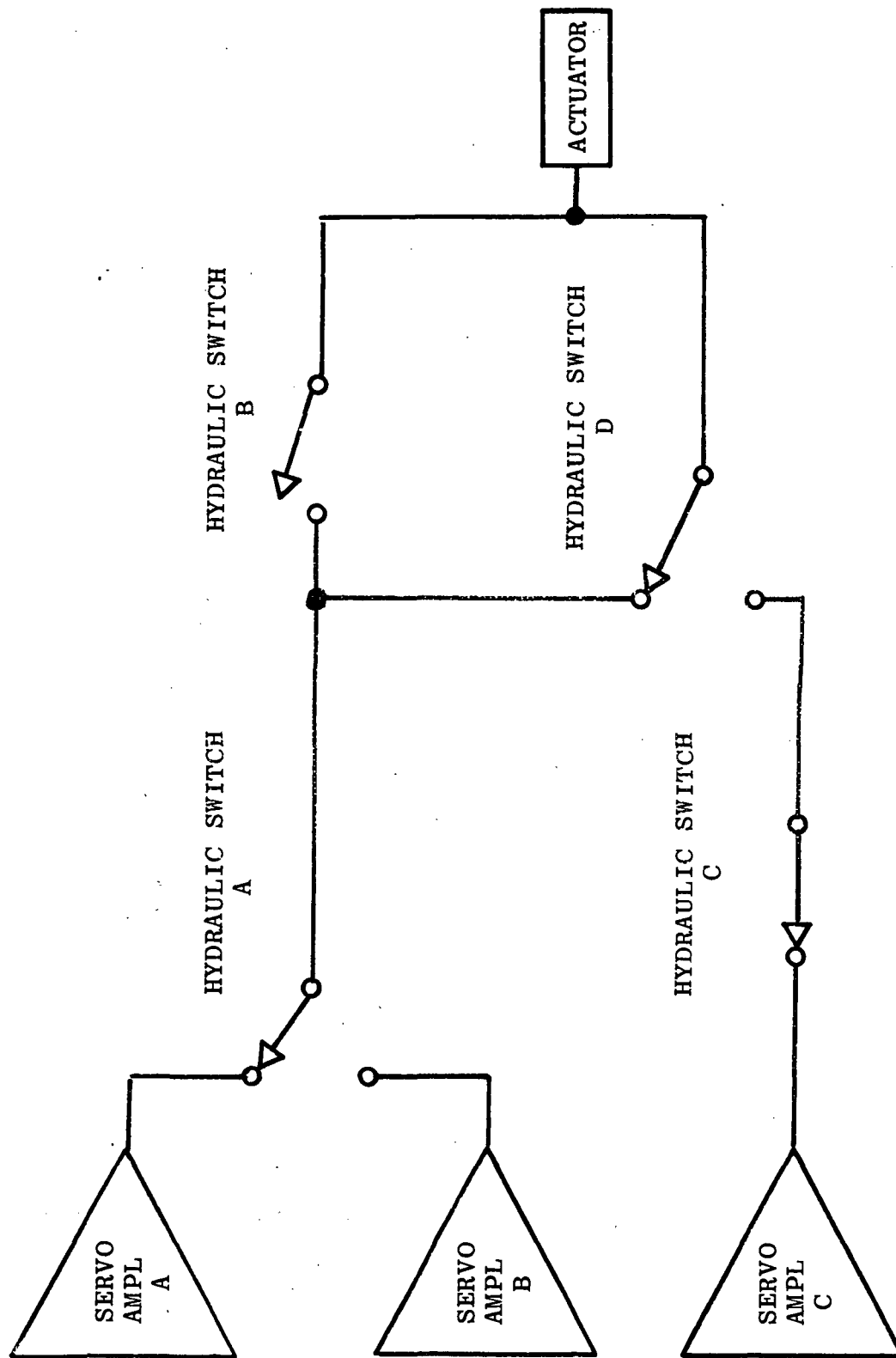
2.1.1 Mathematical Model Functional Description

2.1.1.1 Dedicated Processor

2.1.1.1.1 General Description

The space shuttle processor is a bus organized machine built around 256 programmable registers and employing microprogrammed control. The machine has a minimum of special registers. All data is moved from one of the 256 possible general registers to a special register. The contents of the special registers are manipulated and the result is moved back to a general register. All of the special registers have general register addresses. This allows direct transfers from one special register to another special register.

The machine executes 16 basic commands with many variations. All commands are 16 bits in length and are arranged in a single format. The processor programs, which are known as microprograms, are placed in control storage and



HYDRAULIC VALVE SWITCH CONFIGURATION
ACTUATOR UNIT ALTERNATE "A"

FIGURE 2

thereafter become a part of the machine hardware. The program can be changed by replacing the content of the control storage. The commands control all aspects of the operation of the basic machine and are executed in a single machine clock cycle.

The eight-bit arithmetic/logic unit performs all manipulations of data including; addition, logical AND, logical OR, logical exclusive OR, and one bit left and right shifts. The results of the arithmetical and logic operations are held in one of the four accumulators (A, B, C, or D). The outputs of the accumulators can be placed on the Main-bus for transfer to the Scratch Pad Memory (SPM) or can be routed via an internal bus to the input of the ALU. Each of the accumulators can be loaded from SPM under program control without affecting the ALU. It is possible to shift data in the accumulators from one accumulator to any of the others under program control. This can be accomplished by utilizing the fact that the carry bit receives the bit shifted out of an accumulator and can be shifted into another accumulator. This capability will enhance the machines ability to program for multiplication and division.

2.1.1.1.2 Main Memory

It is not necessary for the processor to have a main memory in the Space Shuttle project. However, the processor is designed so that main memory can be added. All addresses above 9 and below 64_{10} are available for special I/O registers. Any one or more of these addresses could be used for registers that could be assigned to the main memory for data and address.

2.1.1.1.3 Main-Bus

The main-bus is a bi-directional bus that handles all data flow between the scratch pad memory, I/O devices and the special ALU registers in the processor. The bus functions synchronously with the processor. Data placed on the Main-bus by an I/O device must do so under control of the processor. The accumulators in the processor have an independent bus to the ALU. This allows arithmetic and logic operations to be executed between the accumulators in one cycle. Literal data contained in the R-Address Register is capable of being transferred directly to the L-Register on a data path independent of the Main-bus.

2.1.1.1.4 Control Memory

The control memory can be expanded to 4096 words of control storage. Instructions from the Control Store Memory (CSM) are loaded into the instruction execution register in such a way that the next instruction is waiting to be strobed into the register while the current instruction is being executed. This is a modified pipeline effect and is particularly helpful when branching to pages external to the current page. The memory extend register (RE) is loaded in one cycle. This allows the processor to load RE with a new page number then execute an instruction to load the instruction counter (RL) since this instruction was waiting to be executed as described above. This gives the processor the ability to jump from page to page without the necessity for maintaining a common instruction in every page.

2.1.1.1.5 Special Registers

There are 8 special registers in the basic machine. Their descriptions are as follows:

A, B, C, and D Accumulator Registers - These are designated RA, RB, RC, and RD (RX) respectively on the system block diagrams, Figures 2-1 and 2-2. They are each 8-bit registers that serve as operand registers for most of the operate class commands. All arithmetic manipulations take place on the contents of one of these accumulators. Register RD plays a duplicate role. As well as being one of the accumulators, it also serves as the index register (RX).

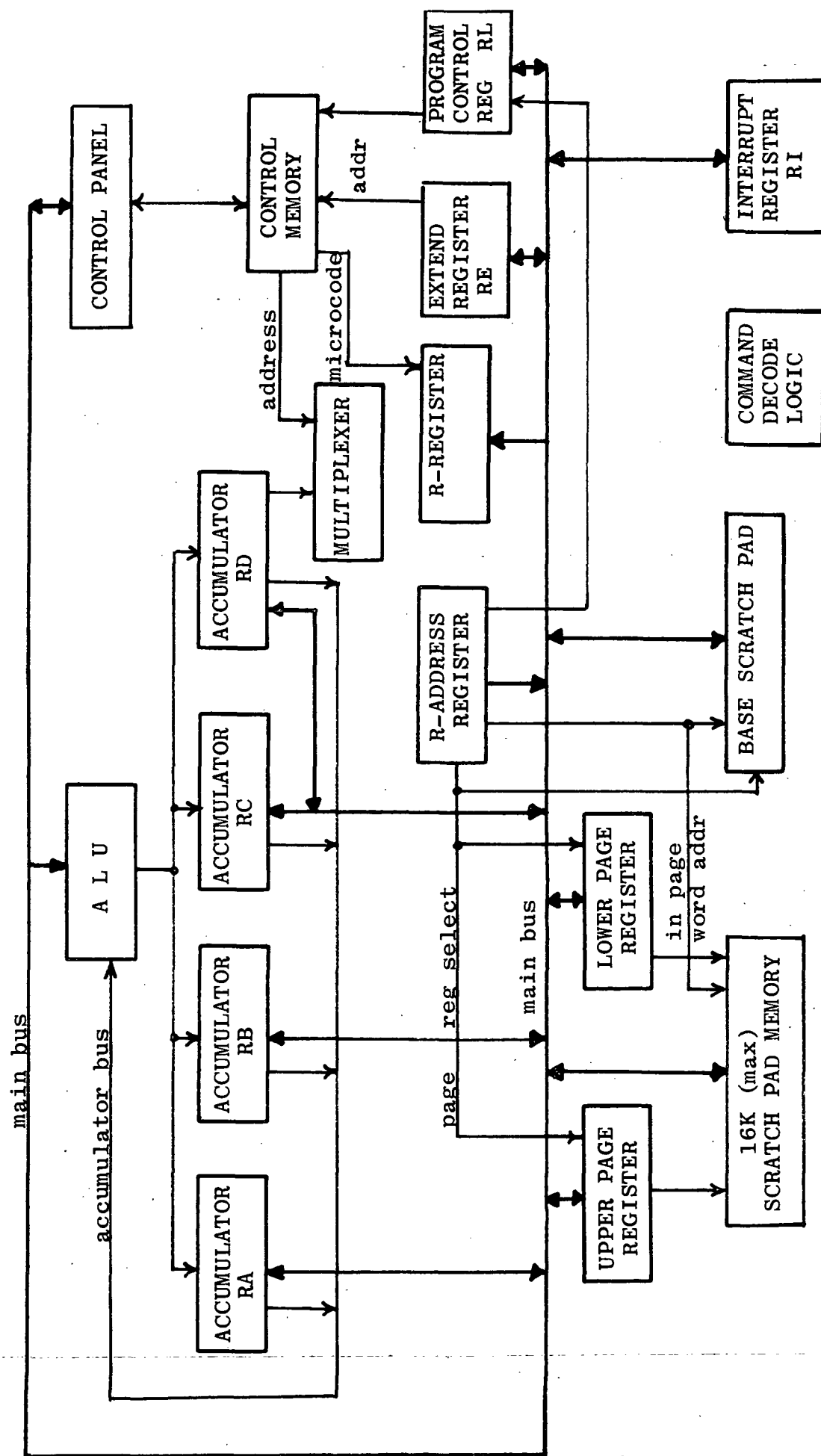
L-Register - This eight bit counter is used as the lower portion of the machines program counter. It is designated RL in Figure 2-1. It contains the control storage address of the next command to be executed, unless altered by a jump or branch command. This register is incremented at each clock time when the processor is running, unless there is a command execution delay imposed.

2.1.1.1.6 Extend Register

The upper four-bits of this register contain the four arithmetic flags. These flags are:

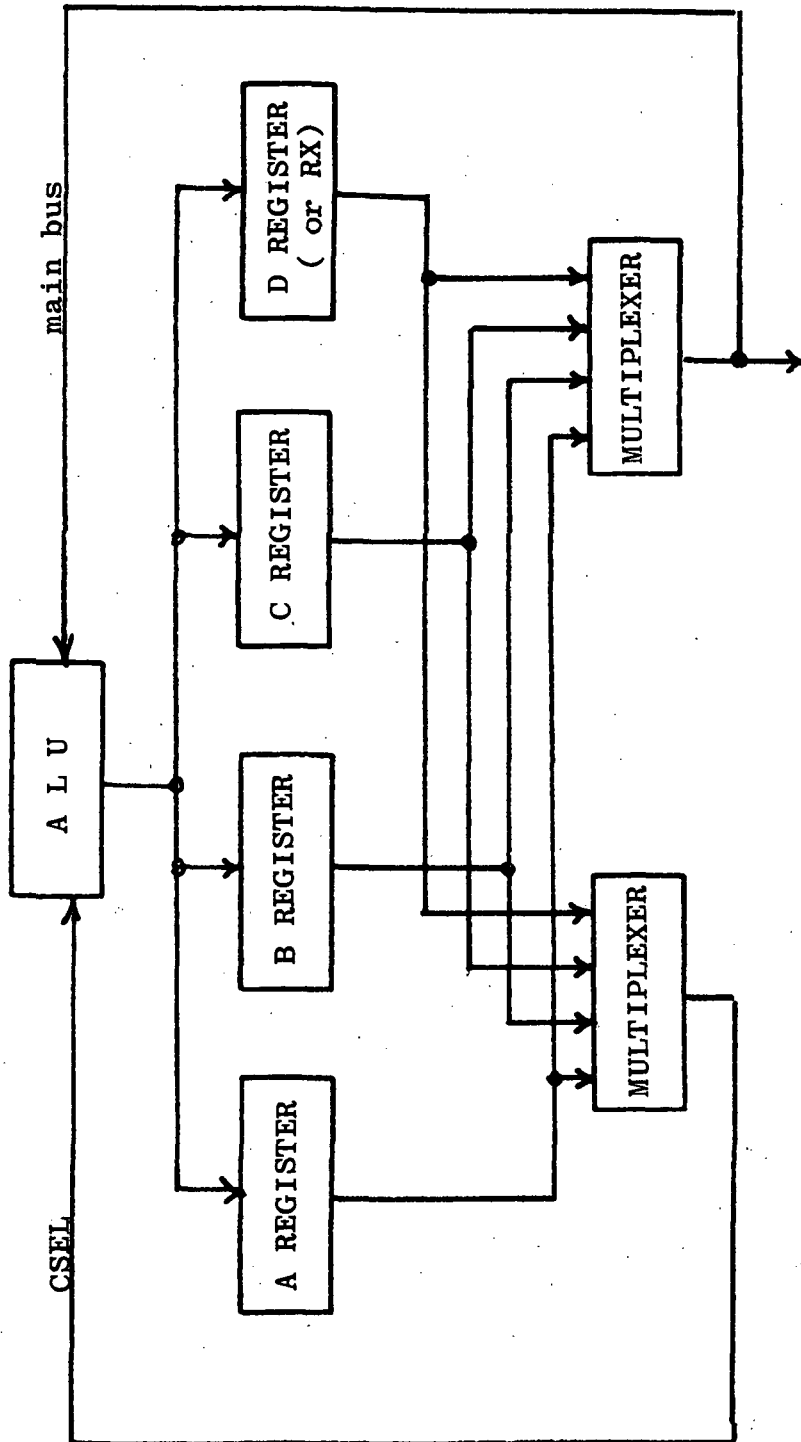
- Z: Zero result
- N: Negative result
- C: Carry
- O: Overflow

Branches can be made on these flag conditions as well as the combination of Z and N to produce a "results positive" conditional branch capability. The



ACTUATOR INTERFACE ALTERNATE "A"
 MATHEMATICAL MODEL PROCESSOR
 BLOCK DIAGRAM

FIGURE 2-1



ARITHMETIC UNIT

FIGURE 2-2

Extend Register is designated RE in Fig. 2-1. The lower four bits of RE are used to extend the address of the control storage.

2.1.1.1.7 Interrupt Register

The interrupt register is utilized as a holding register for the contents of the RL register when an interrupt has been initiated. It is designated RI in Fig. 2-1. When the RI register is stored into the L Register the interrupts are reenabled if they were disabled by a previous command. The interrupt always disables further interrupts.

2.1.1.1.8 Register File

The processor can have up to 16K registers by utilizing full 8-bit page registers to independently reference one of the 256 x 64 word pages. The Scratch Pad (SP) memory is implemented in such a way that paging is over 64, 8-bit words at a time. Each location has a unique address including the special function registers.

They are:

(All on three processor chips)	Loc. 0	A-Register
	Loc. 1	B-Register
	Loc. 2	C-Register
	Loc. 3	Index Register
	Loc. 4	Extend Register
	Loc. 5	L-Register (Program Counter)
	Loc. 6	Page Register 1
	Loc. 7	Interrupt Register
	Loc. 8	Page Register 2
	Locs. 9-63	Reserved for I/O Devices
	Locs. 64-127	Base Page Scratch Pad. It is accessible independent of the page registers.

Addressing the Scratch Pad memory is accomplished by either directly addressing the Base Page (01xxxxxx) or by loading the Scratch Pad Page Registers with the desired page numbers and using (10xxxxxx) or (11xxxxxx) in the ADDRESS bits. This gives the user a 64 word page with the maximum number of pages being determined by the number of bits in each Page Register. Utilizing both page registers will allow access to 192 words of scratch pad memory without reloading the Page registers.

The two 8-bit page registers are multiplexed to allow each to address any of the 16K words of Scratch Pad memory. The system block diagram (Figure 2-1) shows the maximum configuration. The basic processor design allows a great deal of flexibility in register configurations as alternates to those given here.

The four accumulators give the processor the capability of easy manipulation between the accumulators as an aid in programming more complex mathematical functions. It is possible to execute any of the arithmetic functions with any of the four accumulators.

2.1.1.1.9 Processor Interrupts

The Processor Interrupt is brought into the processor on a single line. The effect of initiating an interrupt when they have been enabled in the processor is as follows:

1. The contents of the L-Register are automatically stored in the INTERRUPT Register.
2. The interrupts are automatically inhibited and the interrupt in progress flag is set.

3. An unconditional jump to location 000 is forced.
 - a. Location 000 must contain an instruction to store (save) the contents of the EXTend Register.
 - b. Location 001 must contain an instruction to jump to the interrupt processing routine.
4. When the interrupt processing is complete, the EXTend Register contents are restored and the contents of the INTERRUPT Register are returned to the L Register. This forces a return to the portion of the program being executed when the interrupt was received. It also enables the interrupts.

Setting the interrupt in progress flag allows the processor to test for interrupt condition. This is necessary due to the RESET function.

RESET is an involuntary function that forces a jump to location 000 of the processor base page. This is an error condition which will allow the processor to reinitialize the computer program.

RESET functions in the same manner that normal interrupts do with the exception that the interrupt in progress flag is not set.

2.1.1.2 Microprogramming Language

Refer to the document "Description of Assembly Language for the Programmable Controller" in Appendix F.

2.1.1.3 Mathematical Model Solution

2.1.1.3.1 General Description

The transfer function which describes the hydraulic actuator in mathematical terms is shown in Figure 2-3. In order to make a numerical analysis of this function it was necessary to break the function down into a network of simple first order transfer functions and apply a state variable analysis to the model. See Figure 2-4 for a description of the state variable model for the control system.

$$F = \beta_c \left[\frac{K_v K_L K_T (1 + \tau_2 s) + K_v K_L K_T s [M \tau_2 s^2 + (M + \tau_2 D)s + D]}{M \tau_2 A^2 K_L s^4 + [A^2 K_L (M + D \tau_2) + A K_v H K_T M \tau_2 + M K_T K_L \tau_1] s^3 + \right. \\ \left. [A^2 K_L D + A K_v H K_T (M + D \tau_2) + D K_T K_L \tau_1 + A^2 K_T K_L \tau_2] s^2 + \right. \\ \left. [A H K_v K_T K_L \tau_2 + A K_v H K_T D + A^2 K_T K_L] s + A H K_v K_T K_L \right]$$

$$- \left(\frac{K_c s}{\tau_3 (s+1)} \right) \Delta P_{AM}$$

HYDRAULIC ACTUATOR TRANSFER FUNCTION

FIGURE 2-3

2.1.1.3.2 State Variable Equations

State variables, X_1 through X_7 were assigned as shown in Figure 2-4. In addition, other variables (V_1 , S , Q) were assigned for convenience. The following state equations were written to describe the system:

$$1) \quad \dot{X}_1 = X_2$$

$$2) \quad \dot{X}_2 = \frac{1}{M} \left[K_T (X_3 - X_1) - DX_2 \right]$$

$$3) \quad \dot{X}_3 = \frac{1}{A} [S - V_4]$$

$$4) \quad \dot{X}_4 = V_4$$

$$5) \quad \dot{X}_5 = DPP2S - \Delta P$$

$$6) \quad \dot{X}_6 = \left[\frac{Kc}{\tau_3 \tau_4} \right] X_5 - \left[\frac{\tau_3 + \tau_4}{\tau_3 \tau_4} \right] X_6 - \left[\frac{1}{\tau_3 \tau_4} \right] X_7$$

$$7) \quad \dot{X}_7 = X_6$$

$$\text{where } V_1 = K_T \left[X_3 - X_1 \right]$$

$$V_2 = X_1 + \frac{1}{K_L} V_1$$

$$V_4 = \frac{\tau_1}{\tau_2} \left[\frac{1}{A} V_1 - \frac{1}{\tau_1} X_4 \right]$$

$$DPPLS = \frac{1}{A} V_1$$

$$Q = K_v \left[\beta c - HV_2 \right]$$

$$S = \text{Limit } Q \text{ to } \pm 65$$

The output is given by:

$$\beta_{AC} = V_2 + \dot{X}_6$$

The above equations form a set which can be solved to determine the network response, β_{AC} , to inputs β_C and ΔP .

2.1.1.3.3 Numerical Solution

Once the equations in section 2 have been formulated they must be solved to give an output as a function of time. In this case a sample data solution is desired. That is, the inputs and outputs are sampled and have values only at discrete time intervals, nT , where $1/T$ is the sampling rate and n is an integer. The Runge-Kutta numerical analysis method was selected for finding a solution which could be implemented on an eight bit computer at low sampling rates (See Appendix A for the derivation of the Runge-Kutta Method).

The Runge-Kutta Method is really a family of methods called First Order, Second Order, etc., the order representing the number of times the state equation is solved to obtain one new point in the solution. The Second Order method was selected because of its simplicity as well as providing a stable and reasonably accurate solution at a sample rate of 256 HZ.

As interpreted for this problem, the Second Order Runge-Kutta Method consists of the following. First; estimate a new value of the state vector from the previous value and the previous derivative.

$$\hat{X}(n+1) = X(n) + T \dot{X}(n)$$

Using $\hat{X}(n+1)$, find $\hat{\dot{X}}(n+1)$ from the state equations. Then find the $X(n+1)$ value

$$\text{from} \left(X(n+1) = X(n) + \frac{T}{2} \left[\dot{X}(n) + \hat{\dot{X}}(n+1) \right] \right)$$

2.1.1.3.4 Scaling

The appropriate way to convert the model to fixed point is to first determine the maximum value which any variable can reach, then place the implicit binary point in a position just high enough that the number would not overflow. This was accomplished by evaluating the transfer function on the IBM 1130 using a 16 bit floating point solution. An input step function of 8 degrees to the math model was used to obtain a solution for all network variables versus time from which the maximum values for each variable could be determined. The list of variables and constants with their scaling is shown in Tables 2-1 and 2-2.

2.1.1.3.5 Sampling Rate

Since the processor being used for evaluating the math model does not have a hardware multiply instruction, it is desirable to have a sample rate which is an even power of two. Multiplication by a power of two is merely a shifting operation which consumes much less computer time than a multiply routine. Thus is the reason for choosing 256 HZ (2^8) in lieu of 200 or 250.

2.1.1.3.6 Computer Program

A flow diagram for the computer program is given in Appendix B. The equations listed include the sampling rate and scaling requirements. An example of how the scaling was accomplished is shown in Appendix D. Appendix C contain the computer program listing.

2.1.1.4 Interrupt Processing

The interrupt processing routine processes four types of interrupts with decreasing priority. The interrupt with highest priority is the power fault or power failure interrupt. When this occurs, the computer will execute the

TABLE 2-1
LIST OF VARIABLES

Variable Name	Max. Value	Program Variable	Precision	Binary Exponent (Implicit)	Scale Factor to Actual
X_1	3.82	X (K, 1)	2	-13	$\div 8192$
X_2	19.25	X(K, 2)	1	-2	$\div 4$
X_3	3.82	X(K, 3)	2	-13	$\div 8192$
X_4	.56	X(K, 4)	1	-7	$\div 128$
X_5	200	X(K, 5)	1	+1	X2
X_6	.01	X(K, 6)	1	-13	$\div 8192$
X_7	.001	X(K, 7)	1	-16	$\div 65536$
$\dot{X}_1$	19.25	DX(K, 1)	1	-2	$\div 4$
$\dot{X}_2$	550	DX(K, 2)	1	+3	X8
$\dot{X}_3$	15.3	DX(K, 3)	1	-3	$\div 8$
$\dot{X}_4$	17.9	DX(K, 4)	1	-2	$\div 4$
$\dot{X}_5$	2000	DX(K, 5)	2	-4	$\div 16$
$\dot{X}_6$	.28	DX(K, 6)	1	-8	$\div 256$
$\dot{X}_7$	.01	DX(K, 7)	1	-13	$\div 8192$
V_1	9725	V1	2	-1	$\div 2$
V_2	3.82	V2	1	-5	$\div 32$
V_4	18	V4	2	-8	$\div 256$
ΔP_{Model}	2000	DPPLS	2	-4	$\div 16$
S	65	S	2	-5	$\div 32$
Q	500	Q	2	-5	$\div 32$
β_{Act}	3.82	Y	1	-5	$\div 32$
β_c	8	JW	1	-4	$\div 16$
ΔP_M	2000	JP	2	-4	$\div 16$

<u>Constant</u>	<u>Expression</u>	<u>Value</u>	<u>Program Name</u>	<u>Program Value</u>	<u>Binary Exponent (Implicit)</u>
C_0	K_T	57910	C0	113	9
C_1	$1/K_L$	1.497×10^{-15}	C1	126	-23
C_2	$K_c / \tau_3 \tau_4$	.00241	C2	79	-15
C_3	$\frac{\tau_3 + \tau_4}{\tau_3 \tau_4}$	37.1	C3	74	-1
C_4	$1/(\tau_3 \tau_4)$	343.98	C4	86	+2
C_5	$1/(\tau_2 A)$	.0024	C5	79	-15
C_6	$1/2$	18.85	C6	75	-2
C_7	$1/M$	.062112	C7	64	-10
C_8	D	103.	C8	103	0
C_9	$1/A$	.2	C9	102	-9
K_v limit		65	QLIM	2080	-5
K_v		43.6	KV	87	-1
H		2.092	H	67	-5

TABLE 2-2
LIST OF CONSTANTS

initializing routine which clears the variable values and loads the constant values back into memory. The second interrupt, "Reset", has equal priority with power fault and causes the same actions to take place. However, this is a manual operation initiated by the operator by depressing the "Reset" button on the maintenance Panel.

The third level interrupt is parity. Processing is interrupted when a parity failure occurs. The interrupt routine checks to see if this failure occurred three times in succession. If not, control is returned to the main program at a point nearest the point at which the error occurred and at which no error will occur in the math calculations. Otherwise, the computer will be halted preventing the flag "Computer Good" from being set indicating that the computer has failed.

The fourth level interrupt is caused by the real time clock which occurs 256 times per second. This indicates the time at which another output value is to be calculated.

2.1.1.5 Transfer Function Processing

The method used to model and calculate the transfer function versus time has already been discussed in detail in paragraphs 1 thru 5. The flow chart for this portion of the program is shown in Appendix B, pages B2 thru B4.

2.1.1.6 Comparison Routine

When the processor is first turned on or when a reset or power fault interrupt is caused, a time-out counter is initialized. This counter is sampled and

incremented on every iteration. When the value reaches zero, the first comparison between the math model solution and the actual hydraulic actuator output is allowed to take place. The reason for this delay is to allow time for the math model and hydraulic actuator system to come in sync. Once this occurs, a comparison is made on every sample. The maximum difference allowed is .3 volt. Five consecutive failures must occur before the management routine takes over causing hydraulic valve switches to shuttle and servos deactivated.

2.1.1.7 Hydraulic Management

2.1.1.7.1 General Description

When the math model fails to compare favorably with the hydraulic actuator output, the management routine is entered. First, flags set by the hardware are checked to determine the status of the servo-amplifiers and hydraulic valve switches. From this information the computer management routine can determine what action must be taken in order to get a good system back on line. The management routine will set flags which will tell the hardware what servos must be taken off line, switches that must be shuttled, and which servo will be placed on line.

2.1.1.7.2 Detail Description

The flow chart for the hydraulic management routine is shown in Figure 2-5. Reference should also be made to the Figure showing the hydraulic valve switch configuration (Figure 2). In explaining the management flow

diagram, a particular failure will be assumed and the required solution will be listed. Then the steps taken by the computer program will be listed.

Condition 1 -

Given: Servo-A has failed.

Solution: Disable Servo-A.
Shuttle SW-A placing Servo-B on line.

Step -

1. Check status of SW-A. SW-A has not shuttled.
2. Check status of SW-D. SW-D has not shuttled.
3. Set flag to disable Servo-A.
4. Check to see if Servo-B is good.
5. If Servo-B is valid, exit from management routine and allow time for SW-A to shuttle before comparison takes place again. Next entry point to management routine will be point of exit.

Condition 2 - (Failure occurs in addition to Condition 1)

Given: Servo-A has been disabled.
SW-A did not shuttle after being commanded.

Solution: Shuttle SW-D placing Servo-C on line.

Step -

1. Check status of SW-A. SW-A did not shuttle.
2. Check to see if Servo-C is good.
3. If Servo-C is valid, set flag to shuttle SW-D.
4. Exit from management routine and allow time for SW-D to shuttle before comparison takes place again.

Condition 3 -

Given: Switch D has shuttled.

Solution: Disable Servo-C.
Shuttle SW-C.
Shuttle SW-B placing Servo-A on line.

Step -

1. Check status of SW-A. SW-A has not shuttled.
2. Check status of SW-D. SW-D has shuttled.
3. Set flag to disable Servo-C.
4. Set flag to shuttle Switch C.
5. Check to see if Servo-A is good. Servo-A is valid
6. Set flag to shuttle SW-B.
7. Exit from management routine and allow time for switch B to shuttle before comparison takes place again.

Condition 4 - (Failure occurs in addition to Condition 3)

Given: Switch D has shuttled.
Servo-C is disabled.
Servo-A has failed.

Solution: Disable Servo-A.
Shuttle SW-A placing Servo-B on line.

Step -

1. Check status of SW-B. SW-B has shuttled.
2. Check status of SW-A. SW-A has not shuttled.
3. Check status of SW-D. SW-D has shuttled.
4. Set flag to disable Servo-C.
5. Set flag to disable SW-C.
6. Check to see if Servo-A is good. Servo-A is not valid.
7. Set flag to disable Servo-A.
8. Check to see if Servo-B is good. Servo-B is valid.
9. Set flag to shuttle SW-A.
10. Exit from management routine and allow time for SW-A to shuttle before comparison takes place again.

Condition 5 -

Given: SW-A has shuttled

Solution: Disable Servo-B
Shuttle SW-D placing Servo-C on line.

Step-

1. Check status of SW-A. SW-A has shuttled
2. Check status of SW-D. SW-D has not shuttled
3. Set flag to disable Servo-B
4. Check to see if Servo-C is good. Servo-C is valid.
5. Set flag to shuttle switch D.
6. Exit from management routine and allow time for SW-D to shuttle before next comparison takes place.

Condition 6 -

Given: SW-A has shuttled
SW-D has shuttled

Solution: Disable Servo-C
Shuttle SW-C

Step-

1. Check status of SW-A. SW-A has shuttled.
2. Check status of SW-D. SW-D has shuttled.
3. Set flag to disable Servo-C.
4. Set flag to disable SW-C.
5. Check to see if Servo B is good. Servo-B is valid.
6. Set flag to shuttle SW-A.
7. Set flag to shuttle SW-B placing Servo-B on line.
8. Exit from management routine and allow time for SW-B to shuttle before next comparison takes places.

2.1.1.8 Diagnostic Processing

The diagnostic routine is a computer self-test operation which ensures that the computer is performing all its functions properly.

Bit 4 in I/O word 9 is set every time the diagnostic routine is executed successfully. This tells the hardware that the computer is operational. If this bit is not set, the computer will be disabled. This flag bit will be reset during the following iteration sample.

All the instructions in the computer repertoire are executed and the results are compared with known values. If an instruction fails to execute properly, a bad comparison will result and the computer will halt thus preventing bit 4 in word 9 being set.

In addition, the condition flags are checked-out and the four accumulators are checked to make sure that they perform all arithmetic operations and that every bit in the register changes state.

2.1.1.9 Maintenance Panel

2.1.1.9.1 Purpose of Maintenance Panel

The maintenance panel provides complete control and display facilities. It is primarily used for computer maintenance and system checkout and is not part of the real and final configuration. The panel provides for display and control of the following features:

- . Loading of the paper-tape computer program.
- . Display of the Unibus data

- . Display of output of control memory
- . Display of the control memory address
- . Display of the unibus address
- . Parity fail lights
- . Reset switch for initializing the computer program
- . Allows single step operation
- . Provides means of disabling Real Time Clock
- . Provides for single instruction operation from panel

2.1.1.9.2 Detailed Explanation of Maintenance Panel Switches

RUN

This momentary switch places the computer in the run mode causing it to execute microcommands.

STEP

When this switch is up, the computer executes one cycle when the RUN button is depressed.

RESET DISABLE

This switch prevents a computer reset from taking place.

RESET

This momentary switch generates a reset from the maintenance panel.

SENS 1, SENS 2

When one or both of these switches are up, program control can be altered by testing for this condition.

CSU PANEL

When this switch is up, the computer takes its next instruction from the control storage unit. When the switch is down, the computer takes its next instruction from the maintenance panel.

INPUT PROGRAM AND DISPLAY

When the CSU PANEL switch is down, an instruction can be entered by depressing the appropriate switches.

CLEAR

This switch clears INPUT PROGRAM AND DISPLAY.

OUTPUT

When this switch is up (CSU), the instruction to be executed is displayed on the OUTPUT DISPLAY.

When the switch is down (OP CODE & MODIFIER), the operation code and modifiers of the instruction just executed are displayed in the first eight bits of the OUTPUT DISPLAY. The last eight bits display the unibus data.

REAL TIME INTERRUPT

When this switch is up, the real time interrupt (Real time clock) is inhibited.

PARITY RESET

This momentary push button resets all the parity fail lights.

P1

This light displays any parity fail detected on the Unibus.

P2

This light displays any parity fail detected on the upper half of the output from the control storage unit.

P3

This light displays any parity fail detected on the lower half of the output from the control storage unit.

CONTROL STORAGE UNIT ADDRESS

A twelve bit display indicating the control storage unit address.

BUS ADDRESS

An eight bit display indicating the address on the unibus.

OUTPUT DISPLAY

This readout displays either the output from the control storage unit or the operation code and modifiers of the last executed instruction and the data on the Unibus.

COMPUTER CONTROL (Rotary Switch)

AUTO - Allows automatic read in of the program punched on paper tape.

OFF - Computer and paper tape reader in stand-by.

WORD- Allows one complete word to be read in off the paper tape when the SINGLE push button is depressed.

CHAR - Allows one character to be read in off the paper tape when the SINGLE push button is depressed.

COMPUTER RUN - Control is switched from the paper tape reader to the computer.

TAPE SPEED (FAST, SLOW)

Controls the speed at which the paper tape is read into the computer.

ERROR

When an error is detected while loading from paper tape, the ERROR light will illuminate and the tape reader will stop.

ERROR-RESET

This momentary switch resets the ERROR light.

2.1.1.9.3 Maintenance Panel Operation

2.1.1.9.3.1 Program Loading

1. Initial Panel Switch Set-up
 - a) COMPUTER CONTROL (ROTARY SWITCH) - OFF
 - b) ERROR - IGNORE
 - c) TAPE SPEED - FAST
 - d) OUTPUT - CSU
 - e) SINGLE STEP - UP
 - f) SENS1 - DOWN
 - g) SENS2 - DOWN
 - h) RESET DISABLE - DOWN
 - i) CSU PANEL - UP
 - j) REAL TIME INTERRUPT - INHIBIT
2. Load paper tape in paper tape reader. Forward direction is from right reel to left reel. Be sure the ID information is past the read head.
3. Throw 110 VAC power toggle switch up. Throw +5 VAC power toggle switch up.
4. Set Computer Control Switch to SINGLE WORD.
5. Set tape reader power switch to READER-SPOOLER.
6. Depress ERROR RESET and SINGLE WORD. Tape will load to beginning of tape (BOT).

7. Set ERROR switch to STOP and Computer Control to AUTO.
Program will load in until the end of tape code is reached.
8. Turn off tape reader and set the Computer Control Switch to
COMPUTER RUN.

NOTE: If an error occurs during tape load, the ERROR light will come on and the tape reader will stop. Rewind tape and repeat steps 1-8.

2.1.1.9.3.2 Executing Instructions from Maintenance Panel

1. Initial Panel Switch Set-up

a) COMPUTER CONTROL	- COMPUTER RUN
b) OUTPUT	- OP CODE & MODIFIER
c) SINGLE STEP	- UP
d) SENS1	- DOWN
e) SENS2	- DOWN
f) RESET DISABLE	- DOWN
g) CSU PANEL	- DOWN
h) REAL TIME INTERRUPT	- INHIBIT
2. Depress CLEAR.
3. Enter instruction by depressing Input Program switches.
Input Program Display will show instruction to be executed.
4. Depress RUN. The first 8 bits of the Output Display will show the op code and modifiers of the instruction just executed. The data on the BUS is shown in the last 8 bits of the Output Display. The memory address accessed is indicated by the BUS Address Display.
5. Repeat steps 2-4 for additional instructions.

6. Examples

- a) Examine contents of RB (B accumulator)
 - 1) Clear input program display.
 - 2) Enter 8101 (1000 0001 0000 0001). This instruction loads RB into RB.
 - 3) Depress RUN.
 - 4) Read contents of RB from 2nd half of output display.
- b) Examine contents of location 27 ($1B_{16}$)
 - 1) Clear display.
 - 2) Enter 90 1B (1000 0000 0001 1011). This instruction loads contents of location 27 into the A accumulator.
 - 3) Depress RUN.
 - 4) Reads contents of 27 from output display.
- c) Jump to control memory location 54_{16} .
 - 1) Clear Display.
 - 2) Enter E054.
 - 3) Depress RUN twice. Two program cycles are required for a jump instruction.
 - 4) Control Storage Address lights will show 54_{16} .
 - 5) To display the instruction at location 54_{16} , throw CSU PANEL switch to UP position and OUTPUT switch to CSU. The instruction in control memory location 56_{16} will be displayed on the Output Display.

2.1.1.9.3.3 Program Execution

1. Initial Panel Switch Set-up

- | | |
|---------------------|----------------|
| a) COMPUTER CONTROL | - COMPUTER RUN |
| b) SINGLE STEP | - UP |

- c) SENS1 - DOWN
 - d) SENS2 - DOWN
 - e) RESET DISABLE - DOWN
 - f) CSU PANEL - UP
 - g) REAL TIME INTERRUPT - INHIBIT
2. Depress PARITY RESET
 3. Toggle SINGLE STEP to DOWN position.
 4. Toggle REAL TIME INTERRUPT to DOWN position.
 5. Depress RESET.
 6. Depress RUN.

NOTE: Before resetting TEST PANEL while real time program is running, depress RESET on the Maintenance Panel.

2.1.1.9.3.4 Stop Program Execution

1. Toggle REAL TIME INTERRUPT to INHIBIT
Program stops at end of iteration.
2. Toggle SINGLE STEP to UP position.
3. Contents of memory can now be examined.

2.1.1.10 Paper Tape Generation

The source program written using the minicomputer assembly language is first assembled on the NCR 200 computer which produces a listing of the source program along with the machine code in hexadecimal. An object deck is also punched. The CSU page information is punched in columns 1 and 2; the page displacement is punched in columns 5 and 6 and the instruction code is punched in columns 11 thru 14.

A fortran program was written for the IBM 1130 which reads the object deck and converts the hexadecimal data to paper tape code. A copy of this program is listed in Appendix E along with the card placement for running the program.

Paper Tape Format

Each tape word is made up of ten characters. The first character is a start code indicating the beginning of the word. The next four characters are the control storage unit address giving the location where the instruction is to be loaded. Characters 6 through 9 contain the instruction code and the last character is an end code indicating the end of the word. Figure 2-6 shows a diagram of the tape format.

2.1.2 Simplex Functions

In an effort to reduce complexity and costs in Alternate "A", certain functions were reduced to a simplex format. This was done because their redundant design formats were proven operational both in Alternate "B" and in the APS/DIU Breadboard, a concurrent contract. The functions that were reduced to simplex were:

- (1) Input Data Validity Comparison
- (2) Input Supervisory Validity Comparison
- (3) Status Monitor
- (4) Mathematical Models

Items (1) through (3) involved an extensive use of CMOS with considerable complexity in the discrete architecture. These circuits were configured to be eventually implemented in LSI, which was the cause for their complexity. The designs were proven to be operational, as mentioned, so further duplication is unnecessary. Item (4) was also fabricated simplex because of the TTL architecture. Here, TTL was employed because of the processor speed

P	S	E	M	4	3	2	1	
0	1	0	0	0	0	0	0	
P	0	0	1	x	x	x	x	Hexadecimal Address
P	0	0	1	x	x	x	x	
P	0	0	1	x	x	x	x	
P	0	0	1	x	x	x	x	
P	0	0	1	x	x	x	x	Hexadecimal Instruction
P	0	0	1	x	x	x	x	
P	0	0	1	x	x	x	x	
P	0	0	1	x	x	x	x	
0	0	1	0	0	0	0	0	

 sprocket holes

NOTE: a hole in the tape represents a "1"

P is the parity bit and parity is odd

S is the start bit

E is the end bit

M is the memory designator

PAPER TAPE FORMAT

FIGURE 2-6

requirements where CMOS would perform in the LSI configuration, but not in the discrete and large wire-wrap configuration. Now, three TTL math models, each operating identically, would demand more space and power than is necessary to prove the system concept. The significant difference between the triplex scheme and the simplex scheme is that hardware which would have to be incorporated to "power-down" and "power-up" the backup processors as failures occurred, detected by self diagnostics. The technique for doing this was proven is a similar set of circuits designed for the APS/DIU Firing logic.

2.1.3 Redundant Functions

The circuit functions that remained redundant in order to prove the feasibility of the "look and switch" concept is the servo amplifiers and associated circuits. Here, each servo amplifier is fed to a different valve in the hydraulic system, and the on-line servo is selected by the management routine of the math model processor.

Each servo amplifier has associated with it circuitry which tests the validity of the amplifier. The output of the servo amplifier is fed to an A/D converter, and the digital word is compared to the input word with an 8-bit ± 2 bit comparator. Should the servo amplifier deteriorate outside of the 2 bit boundary assigned, its power is interrupted to take it out of the system. This technique is in addition to the tests run by the math model processor to validate the system.

2.2 Alternate "B" Functional Description

Figure 3 is a simplified block diagram of the Alternate "B" system. As with Alternate "A", there are two information input ports from the Data Terminal: data word and supervisory word; both of which are formulated within a 20-bit

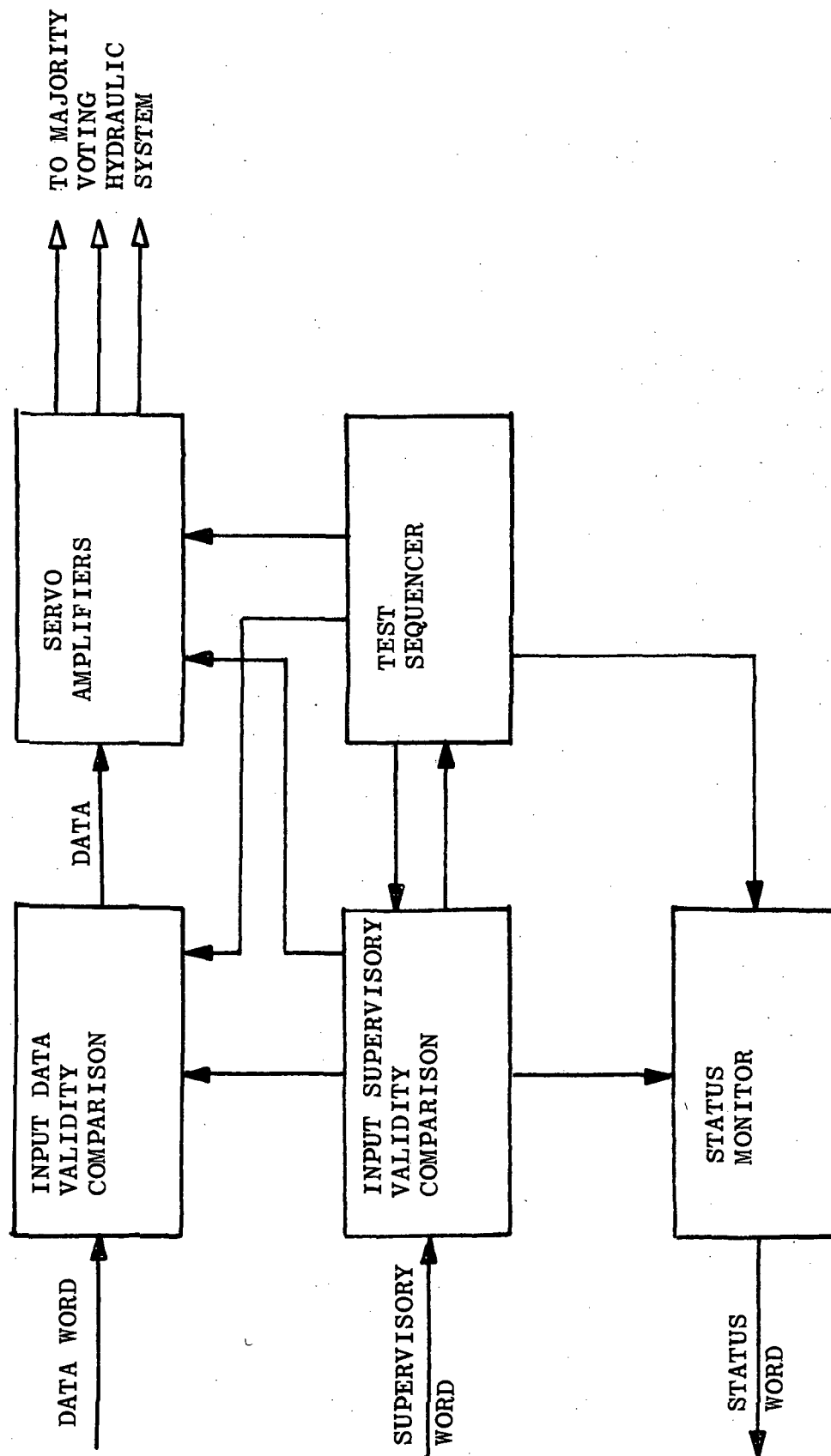


FIGURE 3 ACTUATOR INTERFACE UNIT; ALTERNATE "B"

Polar RZ word frame. Both inputs are, in fact, four-pair inputs all intended to carry identical information (four pair for data and four pair for supervisory). The validity comparison functions perform both serial and parallel voting such that the output information available is immune to any two failures behind it.

It is not required that the circuitry or the Alternate "B" system be dual-failure immune. It is, in fact, a majority-voting system which majority votes two-out-of-three and, hence, is single failure immune. However, because the validity comparison circuitry and the status monitoring circuitry was designed to be implemented in LSI, there is no cost penalty to include these same designs in Alternate "B" even with the additional redundancy. Therefore, these designs were carried over from Alternate "A" and implemented in "B".

The data outputted from the Input Data Validity Comparison function is fed to three separate D/A converters which drive three servo amplifiers. All three amplifiers are on-line in that they feed the majority-voting hydraulic actuator. A failed amplifier will be powered down if its output differs in value from the digital input by more than ± 2 bits. Here, for each amplifier, an A/D converter digitizes the servo amplifier output and this value is compared to the input word.

The commands decoded by the Input Supervisory Validity Comparator are used to control the system configuration as well as extract status information from the Status Monitor. For example, any or all of the servos can be enabled or disabled by command. Or, the status of their operation can be confirmed.

To verify the operational status of the complete system, a Test Sequencer is provided to respond to a supervisory "self-test" command. During this operation, the system ignores any input commands or data until the self-test is completed. Each function of the system is stimulated with appropriate built-in generators, while the redundant sections are inhibited in sequence. The resultant performance is recorded and encoded within a 20-bit status word which is sent to the data terminal at the end of the test. This word reveals the operational status of all of the functions that may not otherwise be noticeable in the normal operation of the system.

2.3 Test Set Functional Description

Figure 4 is a simplified block diagram of the Test Set designed to interface with both the Actuator Interface Units, Alternates "A" and "B" and the APS/DIU breadboard delivered under a separate contract.

The Test Set design and operational philosophy is oriented such that it replaces the Data Terminal which is the normal interface between the subsystem and the data bus system. Input words can be manually or automatically generated and displayed as they are sent to the subsystem. Further, the status words requested and generated by the subsystem can be displayed. The discrete outputs of the subsystem such as servo amplifier outputs, and hydraulic switch status and commands are displayed to represent the termination of the subsystem.

Figure 4-1 is a photograph of the Test Set front panel. Starting at the bottom of the picture, there is a section labeled "MANUAL SUPERVISORY WORD". This section includes the toggles controlling bits 9 through 16 where the 16th bit is labeled "PARITY" either "ODD" or "EVEN". A set of display lamps above the toggles are labeled "SUPERVISORY WORD" and this is the display of the word

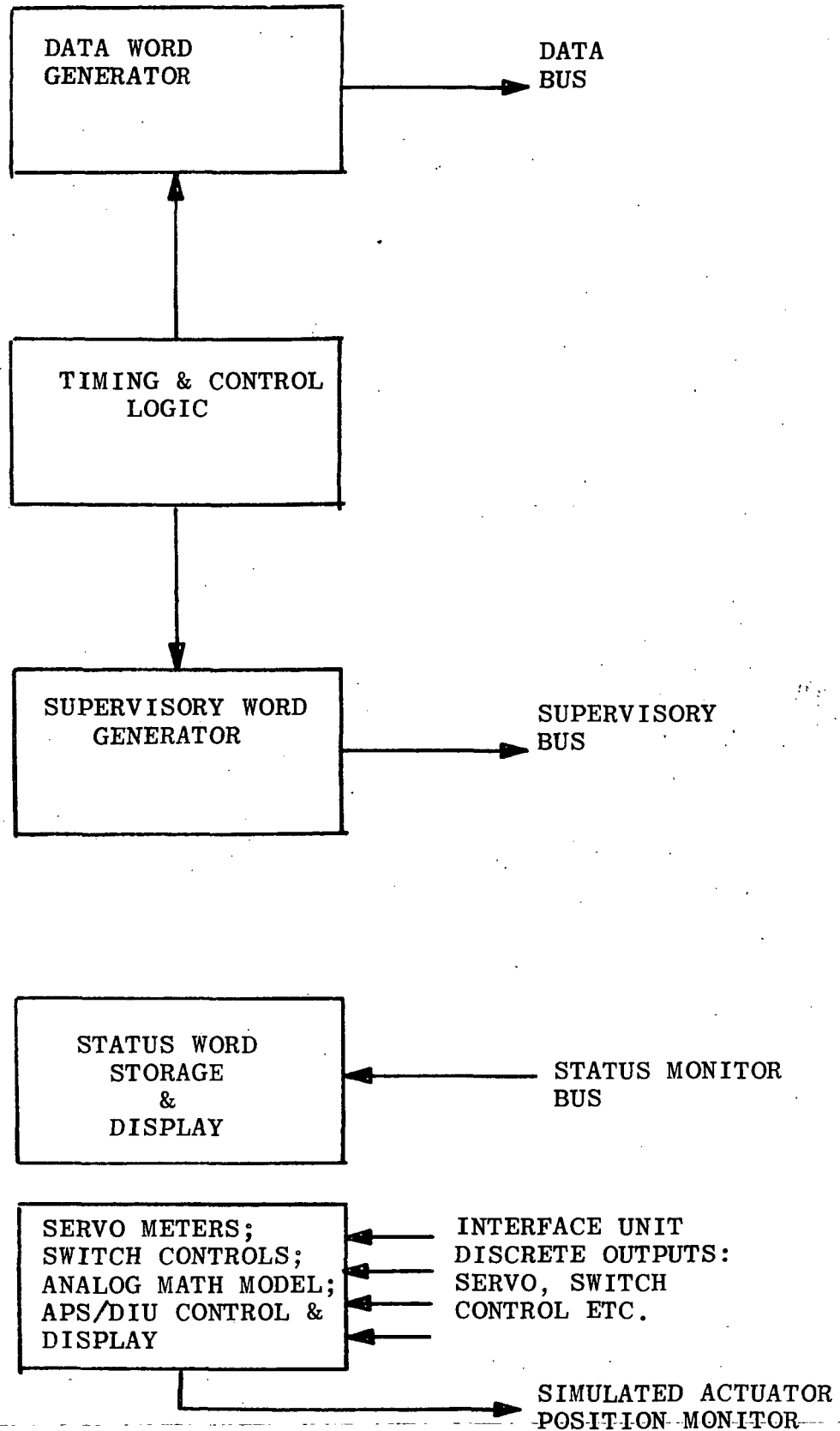
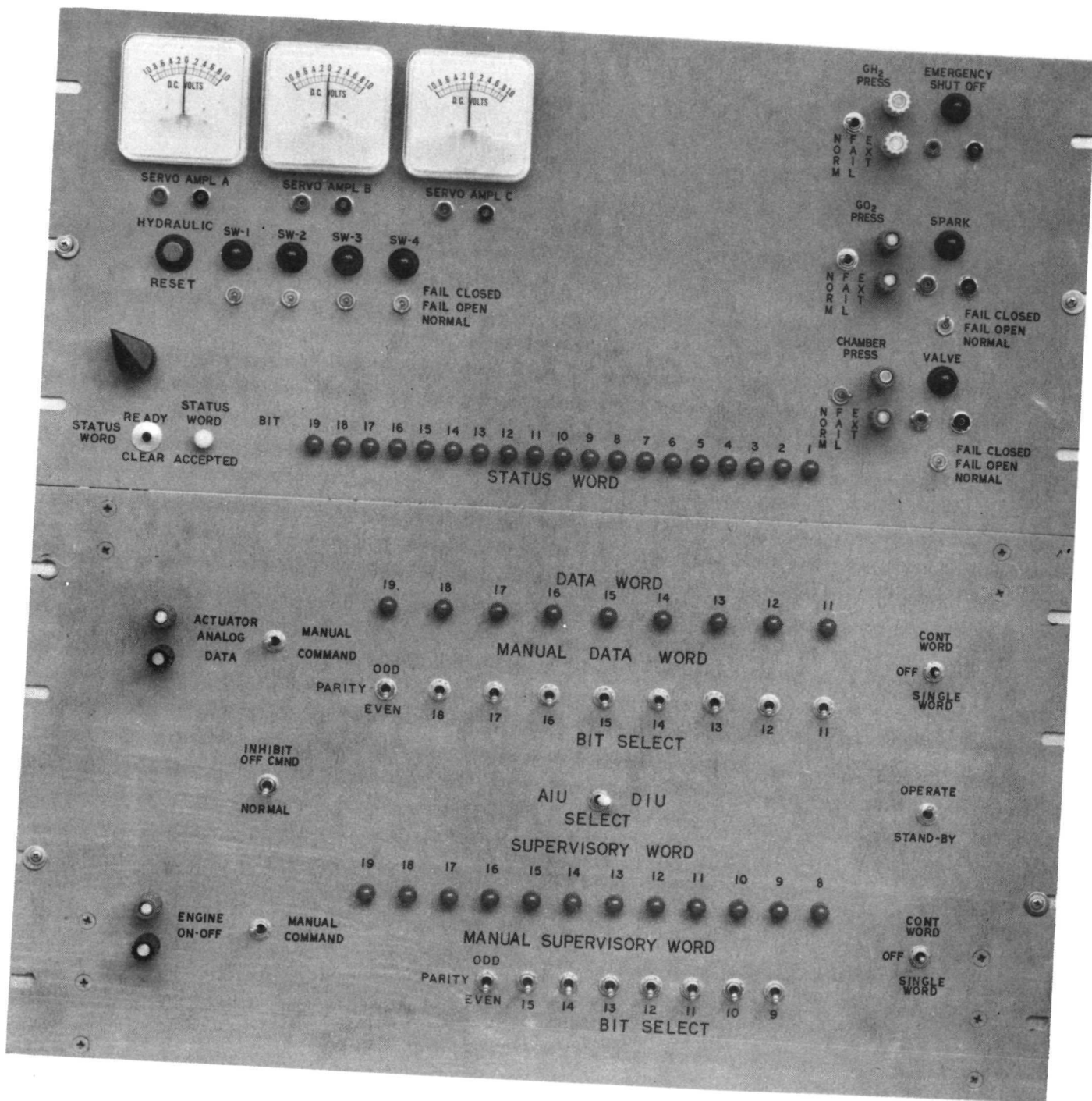


FIGURE 4 TEST SET BLOCK DIAGRAM



ACTUATOR INTERFACE/ APS-DIU TEST SET
FIGURE 4-1

going out on the supervisory bus (or the last word that went out) depending on the state of the toggle to the right of the "MANUAL SUPERVISORY WORD" section. This toggle is labeled "CONT WORD," "OFF," and "SINGLE WORD". In the "CONT WORD" position, the selected word is outputted within each successive word frame. In the "OFF" position, all zero's is sent and in the "SINGLE WORD" position, one word is sent for each depression of the switch into the momentary position. . . on the left side, a toggle is labeled "ENGINE ON-OFF; MANUAL COMMAND" with a pair of binding posts further to the left of the switch. "MANUAL COMMAND" permits the operations just discussed. That is, selecting and sending a word. "ENGINE ON-OFF" is strictly an APS/DIU function where the auxillary propulsion engines are pulsed automatically. Here, a square wave generator is fed into the binding posts and the appropriate supervisory command is formulated and sent with each leading edge.

The top half of the bottom panel is the MANUAL DATA WORD with toggle bit selection and bit display similar to the supervisory section. Again, the display illustrates the words going out (CONT WORD) or it illustrates the last word out (SINGLE WORD). With no appropriate word being transmitted, no information goes out at all. This is different from the supervisory section where no word is an all zero condition. All zero's in data corresponds to a maximum negative excursion of the actuator and is, in fact, a legitimate data word.

To the left of the MANUAL DATA WORD section is a toggle labeled MANUAL COMMAND, ACTUATOR ANALOG DATA. MANUAL COMMAND permits manual selection of data as discussed. The ACTUATOR ANALOG DATA position permits inputting a waveform generator on the adjacent binding posts and automatically updating the continuous data word in accordance with the analog input. An A/D converter digitizes the input and the output data word is updated within every

20usec word frame. Care must be exercised on this input to keep its limits within ± 5 volts. Values greater than 5 volts will cause the A/D converter to overflow and the resultant data sent out will be grossly distorted.

The bottom of the top panel contains the status monitor display labeled as STATUS WORD. All 19 bits are displayed and labeled. A four position selector switch is provided such that the status word may be selected from any one of the four lines outputted from the subsystem. The CLEAR button clears out the display. And, the STATUS WORD ACCEPTED lamp indicates that a complete word has just entered the status display. This lamp is necessary for the confidence of knowing that all-zero word has, in fact, arrived or that the displayed bits are truly in the correct position.

The four lamps below the meters display the activity of the hydraulic switches employed in the hydraulic system for Alternate "A". Supposedly these switches can only be shuttled once. The test set internal relays are, therefore, in a latching configuration. A reset switch is provided to normalize the hydraulic configuration and is labeled as HYDRAULIC RESET. Additional toggles are provided to simulate failures of these switches. These toggles actually alter the state of the status contacts that is monitored by Alternate "A". And, the switches appear to have operated NORMAL, FAILED OPEN or FAILED CLOSED.

The meters are included as simple monitors and are not intended to be accurate or calibrated readings of the servo amplifier outputs. Pin jacks are provided below the meters for accurate readings of the servo amplifier outputs.

The three sets of lamps, binding posts and toggles to the right of the top panel are the output controls and displays of the APS/DIU breadboard and they are not used in the A.I. function.

2.3.1 Test Set System Operation

2.3.1.1 Timing Logic

Figure 5 is the schematic of the AIU/DIU Test Set Timing Logic. This logic formulates the basic word format for the supervisory words and the data words. The 74161's at locations C4 and X4 yield the basic 20 bit frame count. Gate E4, pin 6 provides the supervisory clock which is 20 bits with the 20th bit missing. One-shot D4 inhibits the 20th bit and provides a trigger for the counter reset. Gate E4, pin 8, provides the data clock for bits 11 through 19 when required and as controlled by latch B3, pins 8 and 11. Bit count decoding is by gates B4 and A4. The remaining logic provides the timing and control for the data generator and supervisory generator to recirculate their generated words prior to storage and display. Recirculating insures that the word displayed is, in fact, the word that was transmitted to the subsystem.

2.3.1.2 Supervisory Word Generator

Figure 6 describes the logic to generate the supervisory word. The manual program is entered with the toggles on the left and their outputs are fed to a multiplexer section. The alternate word that is also fed to the multiplexer is the "engine on-off" automatic command which is derived from an external square wave and used for the APS/DIU. The supervisory word selected is loaded into the 9300 shift register and shifted out to the polar RZ driver, 75325 according to the timing logic control. The LM106 provides the line detection to re-enter the information to the shift register prior to loading into the display storage 7475's. It can be seen that the identical information is outputted to the subsystem four pair bus where the bus is divided between two cables.

2.3.1.3 Data Word Generator

Figure 7 is the logic for the Data Word Generator. The configuration here is similar to the supervisory word generator in that the programmed data word is generated by the toggle positions illustrated to the left of the schematic. The switches are fed to a multiplexer that selects either the manual word or the word generated by the internal A/D converter. The selected word is then loaded into the shift register for shifting out through the polar RZ modulator to the subsystem bus. Recirculation and storage loading operates similarly here also.

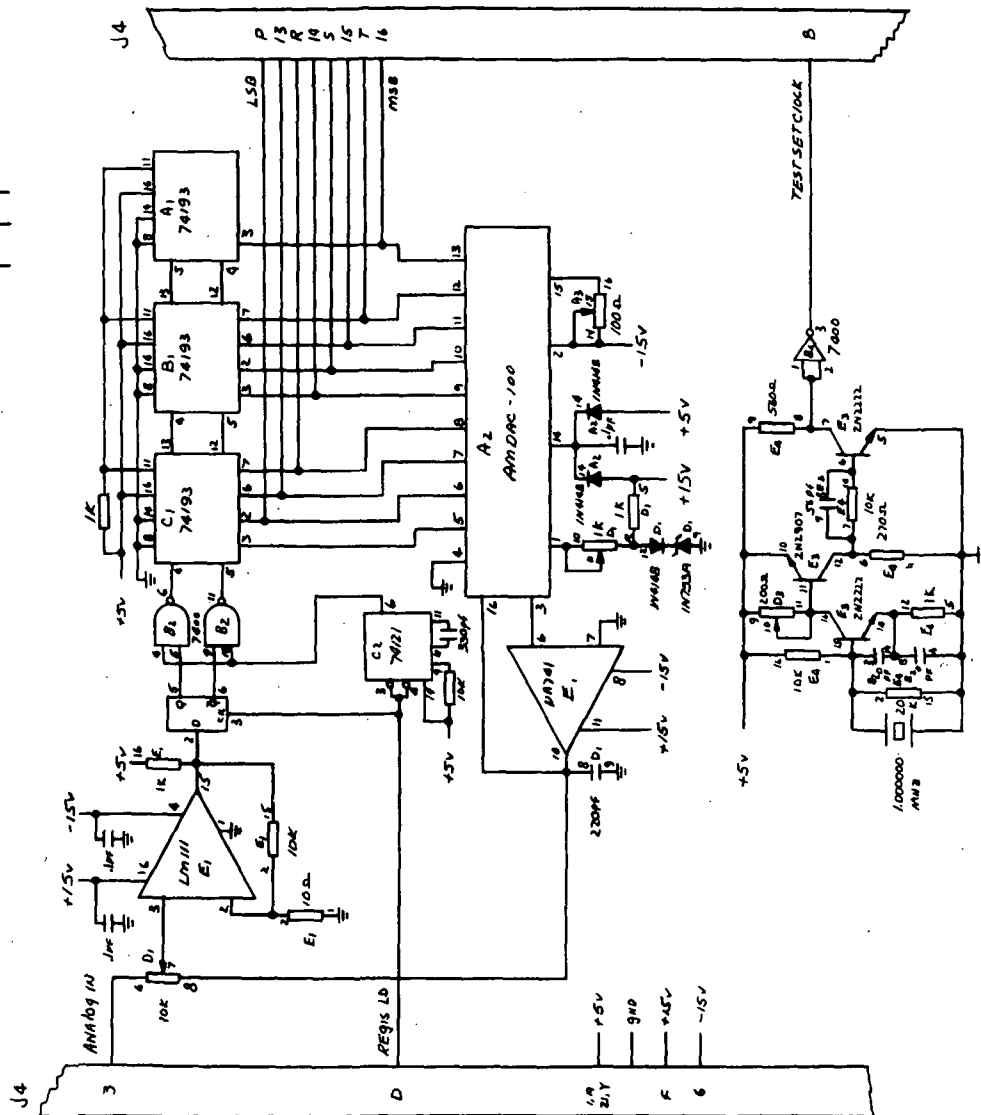
2.3.1.4 Clock Generator and A/D Converter

This function, illustrated in Figure 8, provides the total system 1 MHz clock utilizing a crystal. The A/D converter portion is used to convert any $\pm 5V$ analog input into an 8 bit word updated every 20 usec. The converter is a tracking type where an up/down counter controls an A/D converter in the feedback loop.

2.3.1.5 Display Panel Status Monitor

This section receives and displays the status word received from the subsystem. Figure 9 illustrates the line selector, bit counter, shift register and storage register used to recognize the status word. The bit counter insures that a complete word has been received prior to updating the display. A lamp, L27, is provided to indicate that a new word has been received in the event that the word is an all-zero word. The shift register, storage register and display drivers is a straight-forward design implementation.

REVISIONS		
ZONE	LTR	DESCRIPTION
		DATE
		APPROVED

[illegible]

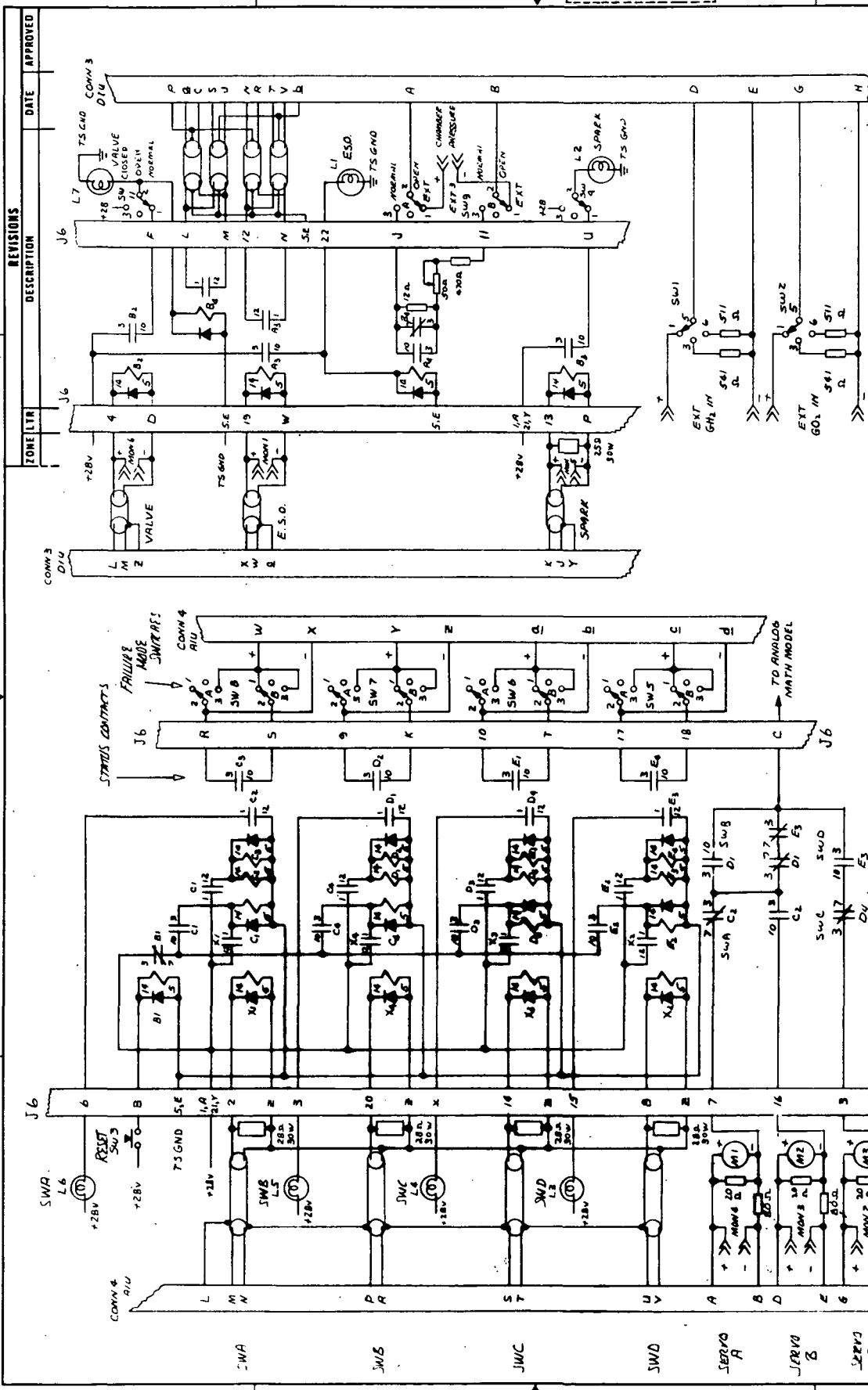
2.3.1.6 Display Panel Switch Logic

Figure 10 is divided up into two basic sections; the right half illustrates the relay logic required for the APS/DIU firing and emergency override valve switch controls and display and the left half illustrates the relay logic required to simulate the Alternate "A" hydraulic system switches and display. The left half will be discussed here.

The basic function of this switch complex is to duplicate as near as possible the response of the hydraulic system to a configuration command. Relays were employed because of their inherent switching delay - an approximation of the delay experienced in shuttling a hydraulic switch. The switch complex provides the following:

- (1) Isolation between the switch drivers in the subsystem and the test set power and ground system
- (2) A lamp load for each switch driver
- (3) A latching characteristic for each switch that has to be manually reset
- (4) Signal path isolation and continuity as would be accomplished in the hydraulic system
- (5) Isolated status contacts for feed back to the subsystem with failure mode control

The relays X1, X4, X3, and X2 represent the load coils for switches A, B, C and D in that order. A 28 ohm resistor is across each relay coil to draw the lamp load required. Because these switches are pulsed, the driven relays have normally open contact which are used to close other relays which are configured to be self latching. These, in turn, control relays that provide the status contacts and the signal path from the appropriate servo amplifiers



REVISIONS		DATE	APPROVED
1	FILE NO. 100	10/10/50	
DESCRIPTION			
ZONE LTR			
ELECTRONIC COMMUNICATIONS, INC.			
ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF NCR			
A1U/DIU TEST SET			
DISPLAY PANEL SWITCH SET			
SIZE CODE IDENT NO			
C 00724			
SCALE			
FIG 10			
SHEET			
UNLESS OTHERWISE SPECIFIED			
DIMENSIONS ARE IN INCHES			
TOLERANCES: .XX ± .02, .XXX ± .010			
HOLE DIA TOLERANCES			
UP +.008, .501 DIA +.010			
THRU +.008, THRU			
500 DIA -.003, 1.000 DIA -.003			
REMOVE BURRS			
BREAK SHARP EDGES .015 ± .010			
MATERIAL			
APPROVED			
MAINT ENGR			
FINISH			
USED ON			
APPLICATION			
NEXT ASSY			

to the math model. It can be seen that the servo's are selected according to the states of relays C2, D1, E3 and D4. These contacts simulate the hydraulic path described in section 2.1.1 of this report. The result is that a valid servo amplifier output is connected to the analog math model which, in itself, simulates the actuator command response.

2.3.1.7 Actuator Analog Math Model

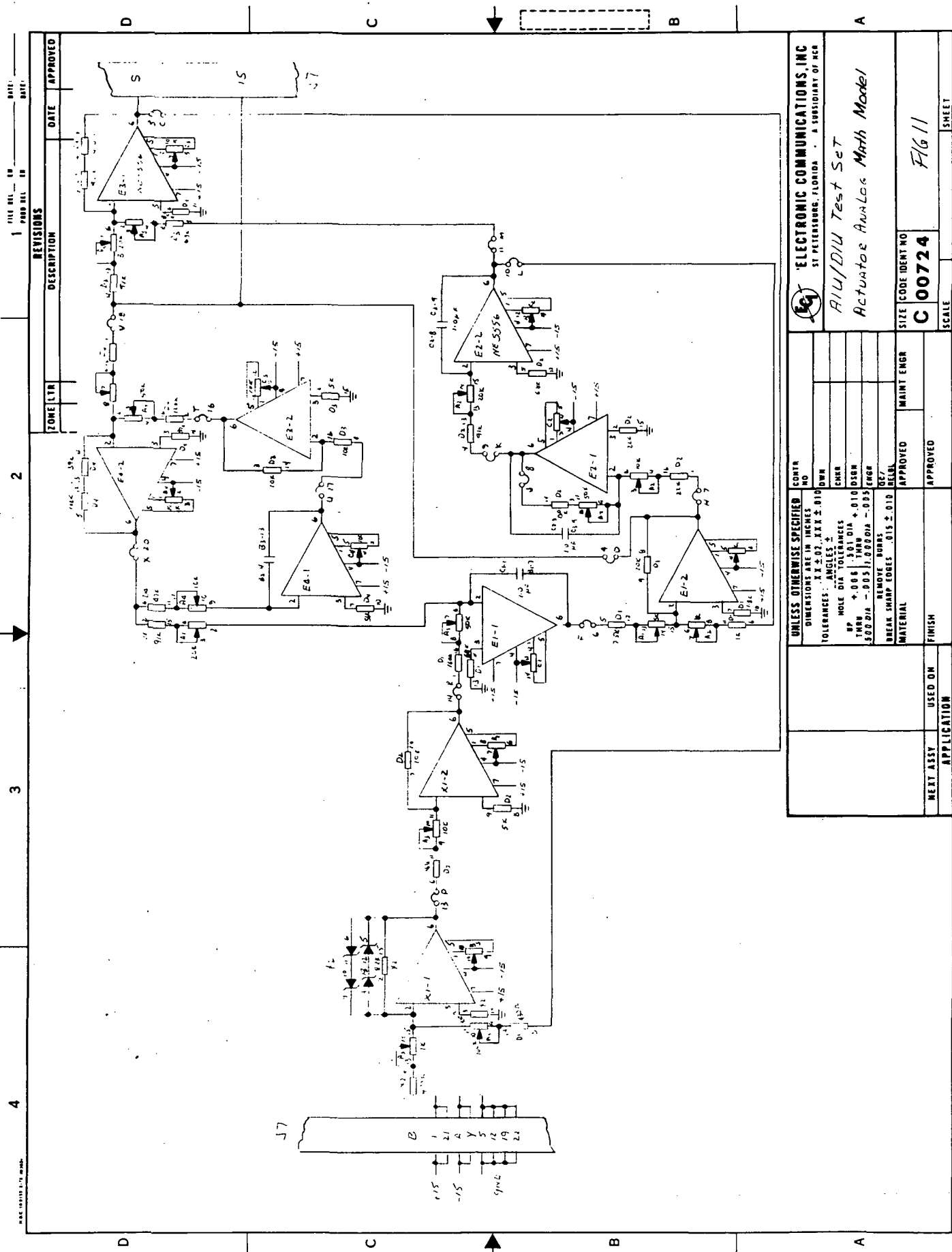
Figure 11 details the hardware required to implement the analog math model. The input to the model is the output of a servo selected by the relay complex just discussed. The outputs of the model represent the actuator position and a delta pressure across the actuator spool. With proper operation, the delta P is zero and always near zero during the transient states.

The analog model utilizes Signetics NE5556 operational amplifiers throughout. Figure 12 is a block diagram describing the transfer function that is implemented in this circuit.

2.3.2 Operational Considerations

Table 2.3 is a listing of the code formats for the supervisory words used to control both Alternate "A" and Alternate "B" Actuator Interface Units.

The normal use of these codes is that they be programmed while the word control toggle is in the "OFF" position. If it is left in the CONT WORD position, unwanted commands will be issued as the desired word is selected. Consequently, operational modes may be commanded that are not desirable or intended.



1 JUL 61 10 000 REL 10

2

3

4

ZONE/LTR	REVISIONS	DATE	APPROVED
	DESCRIPTION		

ELECTRONIC COMMUNICATIONS, INC.
ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF RCA

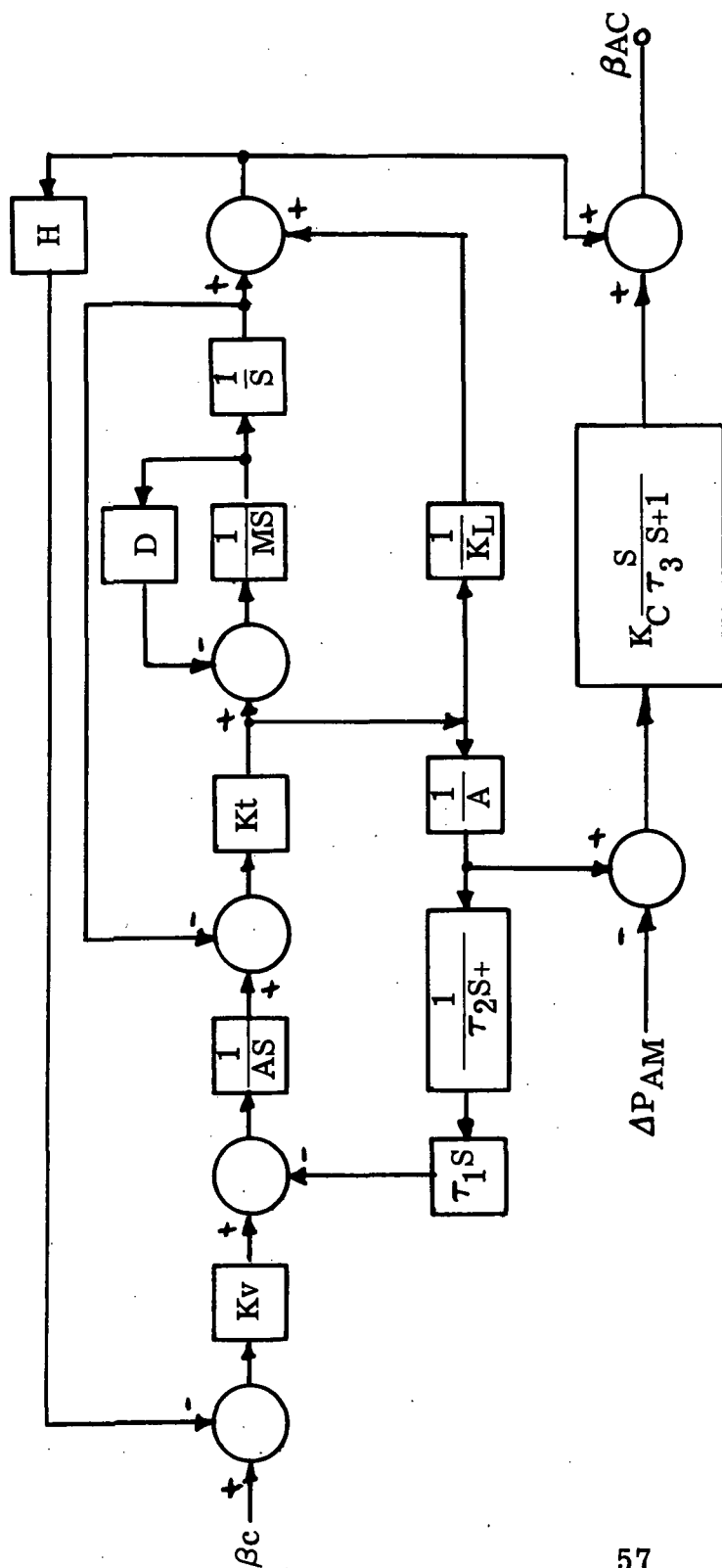
A1U/DIU Test Set
Actuator Analogue Math Model

SIZE CODE IDENT NO
C 00724

SCALE

SHEET

UNLESS OTHERWISE SPECIFIED	CONTR NO
DIMENSIONS ARE IN INCHES	NO
TOLERANCES: ANGLES ±	OWN
HOLE DIA TOLERANCES	CHN
HP	301 DIA
THRU	±.006 IN ±.010 DIA
500 DIA	±.003 IN ±.000 DIA - .005 (ENGR)
REMOVE BURRS	QE7
BREAK SHARP EDGES .015 ±.010 (ENGR)	REL
MATERIAL	
APPROVED	MAINT ENGR
APPROVED	
FINISH	
USED ON	
APPLICATION	



β_c = Actuator Position Command

$$K_v = 43.60 \text{ in}^3/\text{sec/deg.}$$
$$A = 5 \ln^2$$
$$Kt = 57,910 \text{ lb/in}$$
$$M = 16.1 \frac{\text{lb} - \text{sec}^2}{\text{in}}$$
$$D = 103 \text{ lb} - \text{sec}^{\text{in}}$$
$$H = 2.092 \text{ deg/in}$$
$$\Delta P_{AM} = 2.85 \text{ MV/PSI (measured value from actuator)}$$
$$\tau_1 = 6.37 \times 10^{-4} \frac{\text{in}^3/\text{sec}}{\text{lb/in}^2}$$
$$\tau_2 = .05305 \text{ sec}$$
$$\tau_3 = .031 \text{ sec}$$
 $\beta_{\text{ACT}} = 0.5 \text{ V/deg (measured actuator position)}$
$$K_L = 66,800 \text{ lbs/in}$$
$$K_C = .003$$

MATH MODEL

FIGURE 12

TABLE 2.3
ACTUATOR INTERFACE UNIT (S) SUPERVISORY COMMANDS

ALTERNATE "A"

COMMAND CODE BIT POSITION							COMMAND FUNCTION
15	14	13	12	11	10	9	
1	0	0	0	1	1	0	Inhibit Servo Amplifier "A"
0	1	0	0	1	1	0	Inhibit Servo Amplifier "B"
1	1	0	0	1	1	0	Inhibit Servo Amplifier "C"
0	0	1	0	1	1	0	Use Model "A" only
1	0	1	0	1	1	0	Use Model "B" only (these functions deleted)
0	1	1	0	1	1	0	Use Model "C" only
1	1	1	0	1	1	0	(spare)
0	0	0	1	1	1	0	Command Hydraulic Loop 2
1	0	0	1	1	1	0	Command Hydraulic Loop 3
0	1	0	1	1	1	0	(spare)
1	1	0	1	1	1	0	Reset Override (Alternate "A")
0	0	1	1	1	1	0	Rqst Test Pattern 1-19: (1000100010001000100)
1	0	1	1	1	1	0	Rqst Servo Valve 1 current; Input to Servo's MSB-2-0 LSB; MSB 10-17 LSB respectively
0	1	1	1	1	1	0	Rqst Servo Valve 2 current; pressure mon 1 MSB 2-9 LSB; MSB 10-17 LSB respectively
1	1	1	1	1	1	0	Rqst Servo Valve 3 current; position mon 1 MSB 2-9 LSB; MSB 10-17 LSB respectively
0	0	0	0	0	0	1	Pressure monitor 2; Position monitor 2 MSB 2-9 LSB; MSB 10-17 LSB respectively
1	0	0	0	0	0	1	Pressure monitor 3; Position monitor 3 MSB 2-9 LSB; MSB 10-17 LSB respectively

ALTERNATE "B"

1	0	1	0	0	1	0	Inhibit Servo A (vote B & C)
0	1	1	0	0	1	0	Inhibit Servo B (vote A & C)
1	1	1	0	0	1	0	Inhibit Servo C (vote A & B)
0	0	0	1	0	1	0	Reset Override (Alternate "B")
1	0	0	1	0	1	0	Rqst Test Pattern 1-19: (1000100010001000100)
0	1	0	1	0	1	0	Rqst Servo Valve 1 current; Position mon 1 MSB 2-9 LSB; MSB 10-17 LSB respectively
1	1	0	1	0	1	0	Rqst Servo Valve 2 current; Position mon 2 MSB 2-9 LSB; MSB 10-17 LSB respectively
0	0	1	1	0	1	0	Rqst Servo Valve 3 current; Input to Servos MSB 2-9 LSB; MSB 10-17 LSB respectively
1	0	1	1	0	1	0	Rqst Status Power Control Servos A, B & C Bits 2, 3 and 4 respectively; Position mon 3 MSB 10-17 LSB respectively
0	1	1	1	0	1	0	Exercise pre-programmed self-test and report results

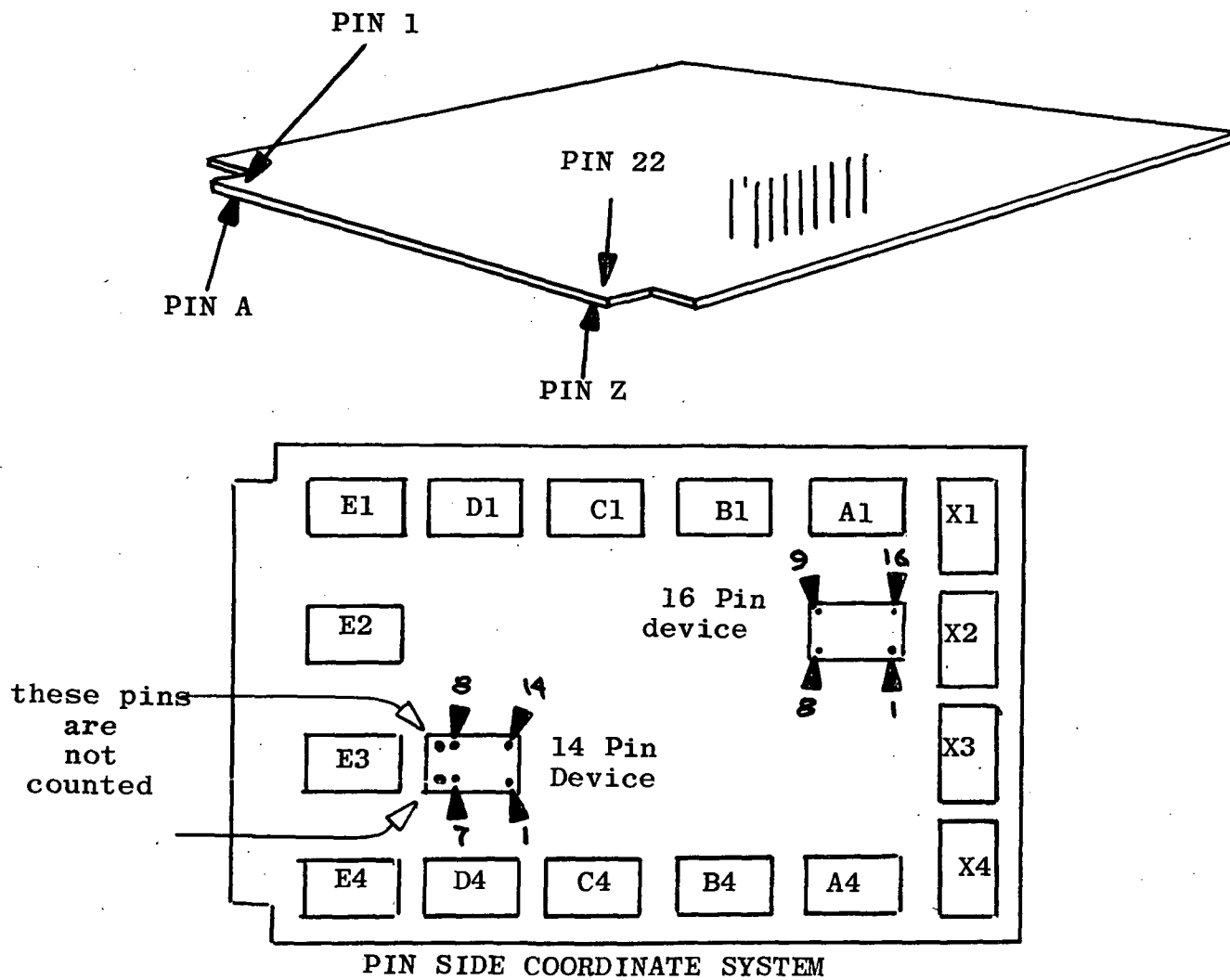
Although the Actuator Interface Test Set is not configured to operate both Alternates "A" and "B" simultaneously, it can be seen that the coding of the supervisory words is not common to both units. A bussed breadboard configuration could be constructed to common the two units.

3.0 SYSTEMS ELECTRICAL DESIGN

Figure A in the appendix of this report is the Systems Block Diagram of Alternate "A". Figures B thru F comprise the Systems Block Diagram of Alternate "B". The designs discussed in this section will be of the modular circuit functions employed in these block diagrams. In some cases, the functional modules are employed in both systems such as the servo amplifiers. They will be discussed only once.

The physical architectures of these breadboarded systems was oriented to yield the maximum in flexibility, modularity and ease of investigation and testing. Circuit functions were cropped and constructed on cards of the type shown in Figure 12-1. The coordinate system assigned to these cards is shown with the basic I.C. layout "A" through "E" and "1" through "4". The "X" coordinates were used only if the board required the additional complexity.

The boards selected yielded sufficient flexibility for installing discrete components as well as I.C. sockets with wire wrap. Each circuit card is assigned a card number for identification against the block diagram and it is assigned a rack coordinate number to identify its location within the system rack. Card extenders are provided with the system such that the cards may be operated external to the framework and tested individually. Care should be exercised in this operation since the card on an extender



CIRCUIT MODULE COORDINATE SYSTEM

FIGURE 12-1

is a rather fragile configuration.

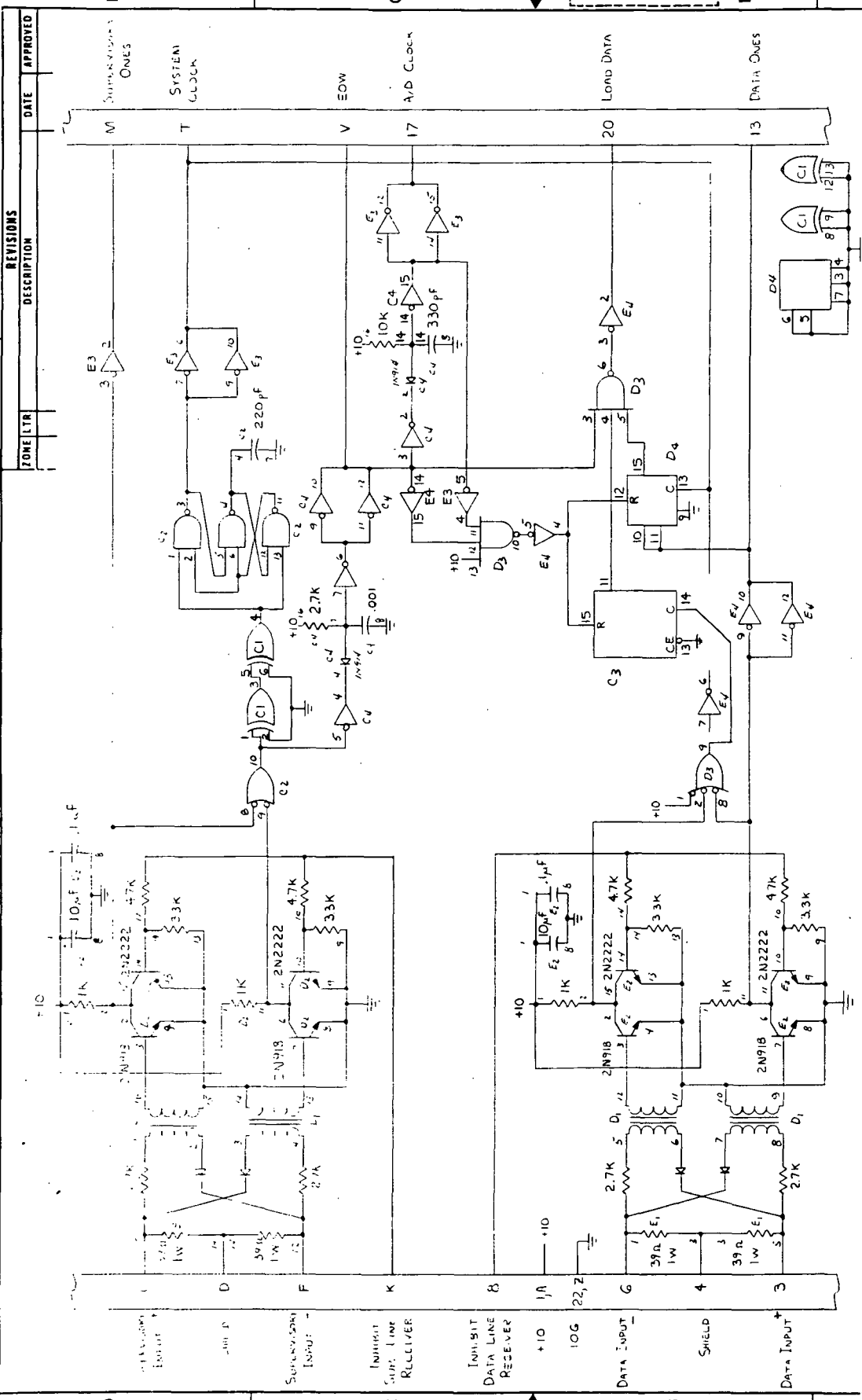
Some cards have auxillary connectors added adjacent to the main board connector with a mating card cage connector in the system. The purpose of these auxillary connectors is to add in/out capability to the card. All connector pin assignments and device pin assignments are on the module schematics. Although the I.C. sockets are all 16 pin sockets, 14 pin devices may be installed. In this case, pins 8 and 9 are ignored for the 16 pin count and the assignment is aligned as if the socket is a 14 pin socket.

3.1 Alternate "A" Circuit Modules

3.1.1 Line Receiver; Clock & Data Bit Counter/Module No. 148; Loc B1

This module accepts both the supervisory words and data words from the data terminal or Test Set busses. Because the Alternate "A" system was reduced to simplex as discussed previously, the circuitry of this module is not redundant. See Figure 13 .

The differential input lines are terminated in 75 ohms and the signal is isolated by a transformer and coupled into transistor data recovery circuits. The system clock is derived from the supervisory input and it is used to clock both the supervisory and data words into the system. A retriggeable one-shot configuration located at C4 detects the missing 20th bit and generates an end-of-word pulse used for leading the accumulated information. This same EOW pulse is shaped and used as a clock source for the A/D converters within the system.



REVISIONS ZONE LTR DESCRIPTION DATE APPROVED		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES: .XX ±.02, .XXX ±.010 ANGLES: .XXX ±.010 HOLE DIA TOLERANCES UP +.008, .501 DIA +.010 THRU +.008, .501 DIA -.001 BREAK SHARP EDGES .015 ±.010 MATERIAL	
CONTR NO 00724		CONTR NO 00724	
SIZE (CODE IDENT NO) C 00724		SIZE (CODE IDENT NO) C 00724	
SCALE 1:1		SCALE 1:1	
FIG 13		FIG 13	
ALU ALT "A" BREADBOARD LINE RECEIVERS, CLOCKS, AND DATA BIT COUNTER (148)		ALU ALT "A" BREADBOARD LINE RECEIVERS, CLOCKS, AND DATA BIT COUNTER (148)	
ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF NCR		ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF NCR	
CD4011 CD4017 CD4009 BOURNE 4252-0011 CD4023 CD4027		CD4011 CD4017 CD4009 BOURNE 4252-0011 CD4023 CD4027	
C2 C3 C4, E3, E4 D1 D3 D4		C2 C3 C4, E3, E4 D1 D3 D4	
DATA INPUT DATA LINE RECEIVER INITIAL SUPPLY INITIAL SUPPLY RECEIVER SYSTEM CLOCK SYSTEM ONES		DATA INPUT DATA LINE RECEIVER INITIAL SUPPLY INITIAL SUPPLY RECEIVER SYSTEM CLOCK SYSTEM ONES	

In the data section, a counter is provided to verify that the proper number of data bits has been received (nine; 8 data plus parity) before permitting the accumulated data to be stored. A serial parity checker at D4 verifies proper parity also before the data can be stored.

3.1.2 One's Supervisory Serial & Storage Registers/Module No. 6; Loc B2

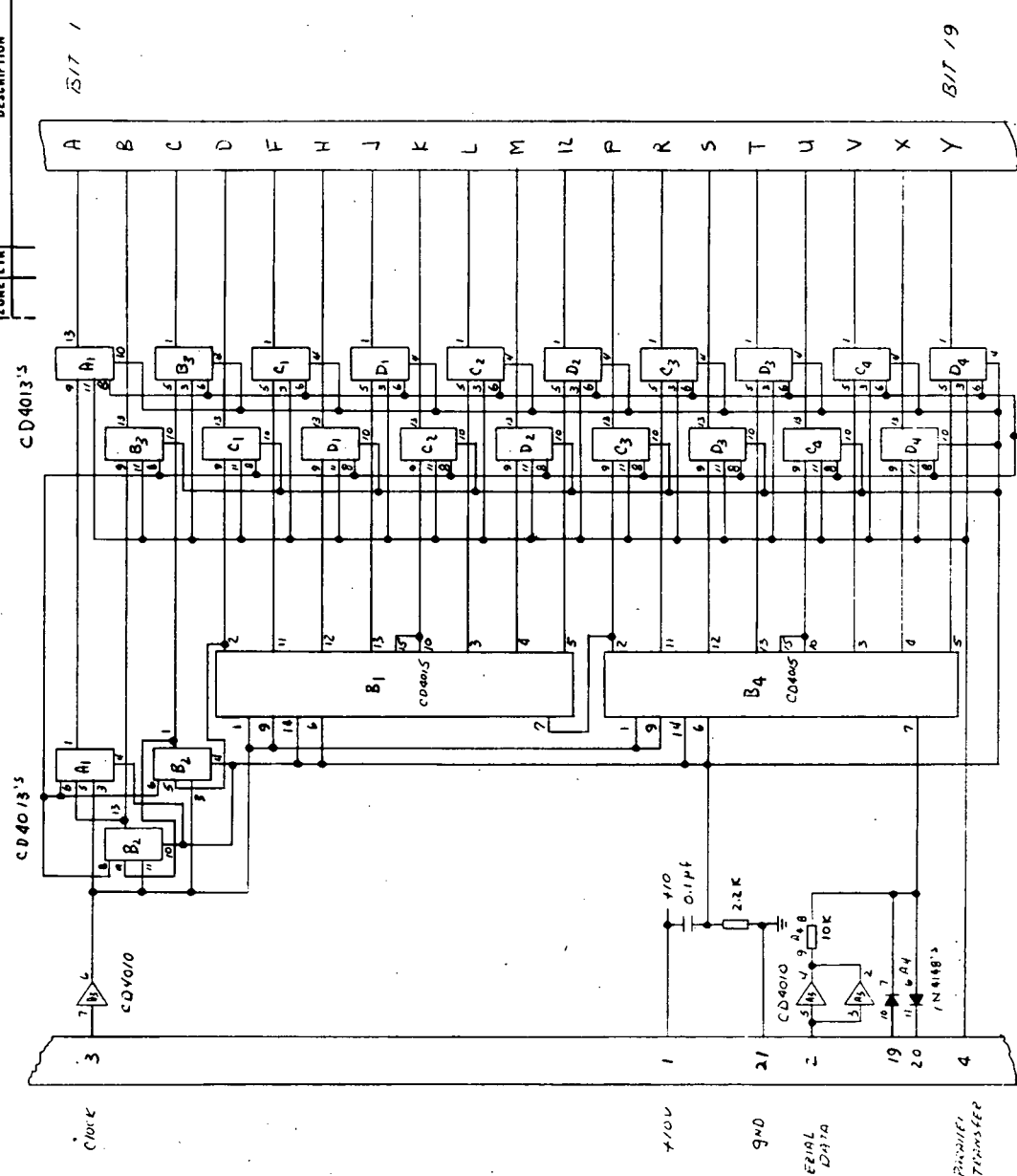
The recovered data from module no. 148 is fed to the One's Supervisory Serial and Storage Registers Module for storage and parallel out-putting to the command decoder. The data inputs on pin 2 and is buffered by gates A3. The diode inputs at pins 19 and 20 are for simulated data generated by the Test Sequencer employed in Alternate "B" only. The supervisory word serially shifts into two shift registers B4 and B1 respectively and is loaded into the CD4013 storage elements at the EOW time. See Figure 14.

3.1.3 Supervisory Decoder & Parity Checker/Module No. 11; Loc B3

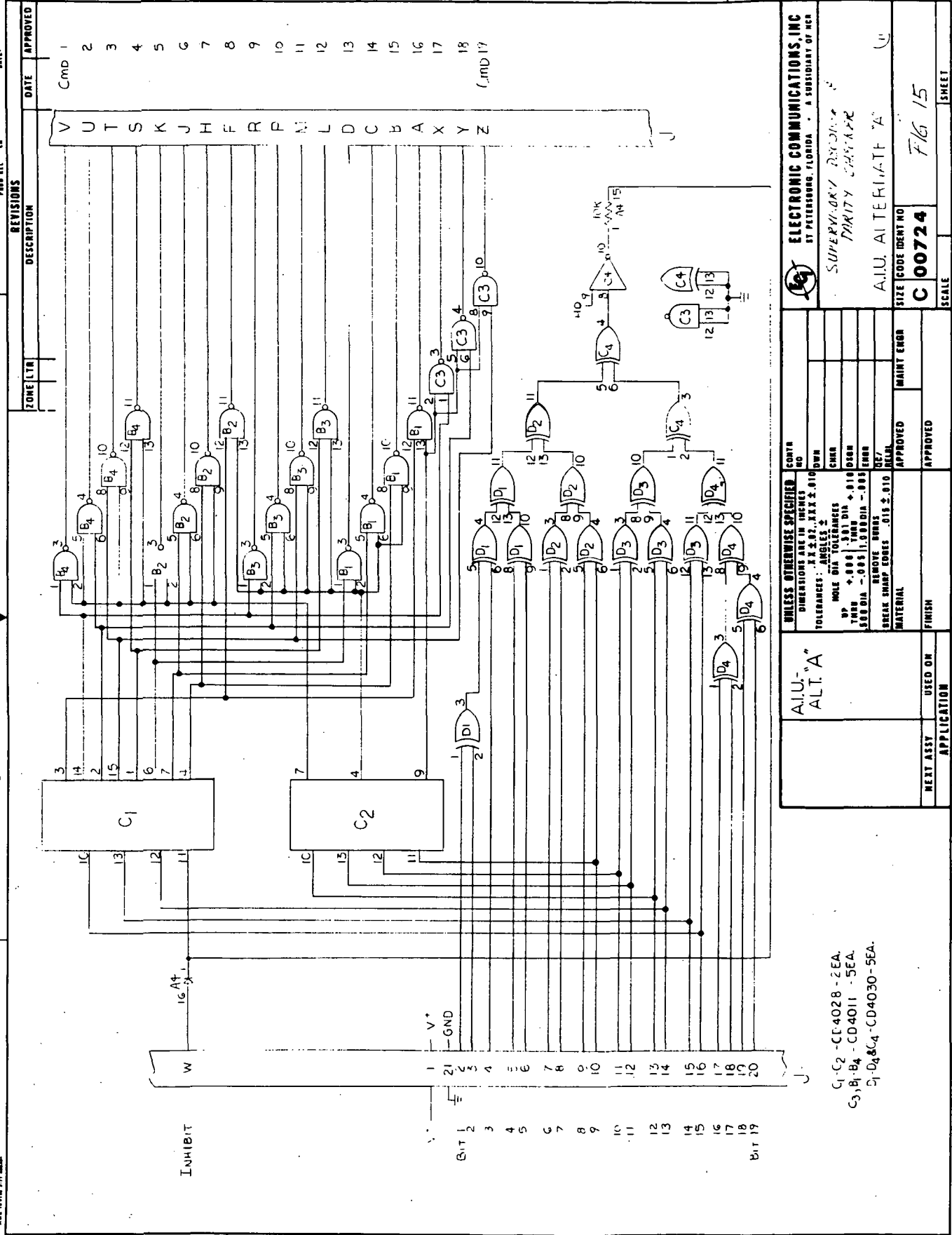
The stored supervisory word is entered into this module in parallel and is decoded by the BCD-to-Decimal decoders C1 and C2 and the accompanying gate logic (See Figure 15). In addition, the parity of the word is verified by a straight forward exclusive-or parity tree and valid parity permits the command to be decoded. This method of parity checking is more complicated than a serial check done on the incoming word. However, it checks the validity of the word in storage which verifies the performance of additional hardware as well as the word itself.

The INHIBIT input on pin W is a Test Sequencer function for Alternate "B" only.

REVISIONS		
ZONE	LTR	DESCRIPTION
		DATE
		APPROVED



UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES: $XX \pm .02$, $XXX \pm .010$ ANGLES: $\pm$	CORN NO DOWN	 ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF NEC	ACTUATOR INTERFACE ALT 'A' MODULE NO. 6 1'S SUPERVISORY SERIAL STORAGE REGIS'rs	SIZE CODE IDENT NO C 00724	FIG 14	SCALE	SHEET
THRU $\pm .006$.501 DIA $\pm .010$ TAPER .500 DIA $-.005$ 1.000 DIA $-.005$	CHSR OSBH ENGR DE/	MAINT ENGR	APPROVED	APPROVED	FINISH	USED ON	NEXT ASSY APPLICATION
REMOTE BURNS BREAK SHARP EDGES .015 $\pm$.010	BLU/	APPROVED	APPROVED	FINISH	USED ON	NEXT ASSY APPLICATION	MATERIAL



REVISIONS		DATE		APPROVED	
ZONE	LTN	DESCRIPTION	DATE	APPROVED	DATE
1	1	111-1111	11/1/64		

UNLESS OTHERWISE SPECIFIED		CONV	
DIMENSIONS ARE IN INCHES		NO	
TOLERANCES: .XX ± .02, .XXX ± .010		OWN	
ANGLES: 2		CHN	
HOLE DIA TOLERANCES		CHN	
TOP +.004, BOT DIA +.010		CHN	
SIDE DIA -.005, 1.00 DIA -.005		CHN	
BREAK SHARP EDGES .015 ± .010		CHN	
MATERIAL		CHN	
FINISH		CHN	
NEXT ASSY		CHN	
APPLICATION		CHN	
USED ON		CHN	
APPROVED		CHN	
MAINT ENGR		CHN	
APPROVED		CHN	
SIZE CODE DENT NO		CHN	
C 00724		CHN	
SCALE		CHN	
SHEET		CHN	

ELECTRONIC COMMUNICATIONS, INC.	
A SUBSIDIARY OF NCR	
SUPERVISORY DESIGNER	
PARITY CHECKER	
A.I.U. ALTERNATE 'A'	
FIG 15	

C1 - C2 - CD4028 - 2EA.
 C3, B1, B4 - CD4011 - 5EA.
 D1, D2, C4 - CD4030 - 5EA.

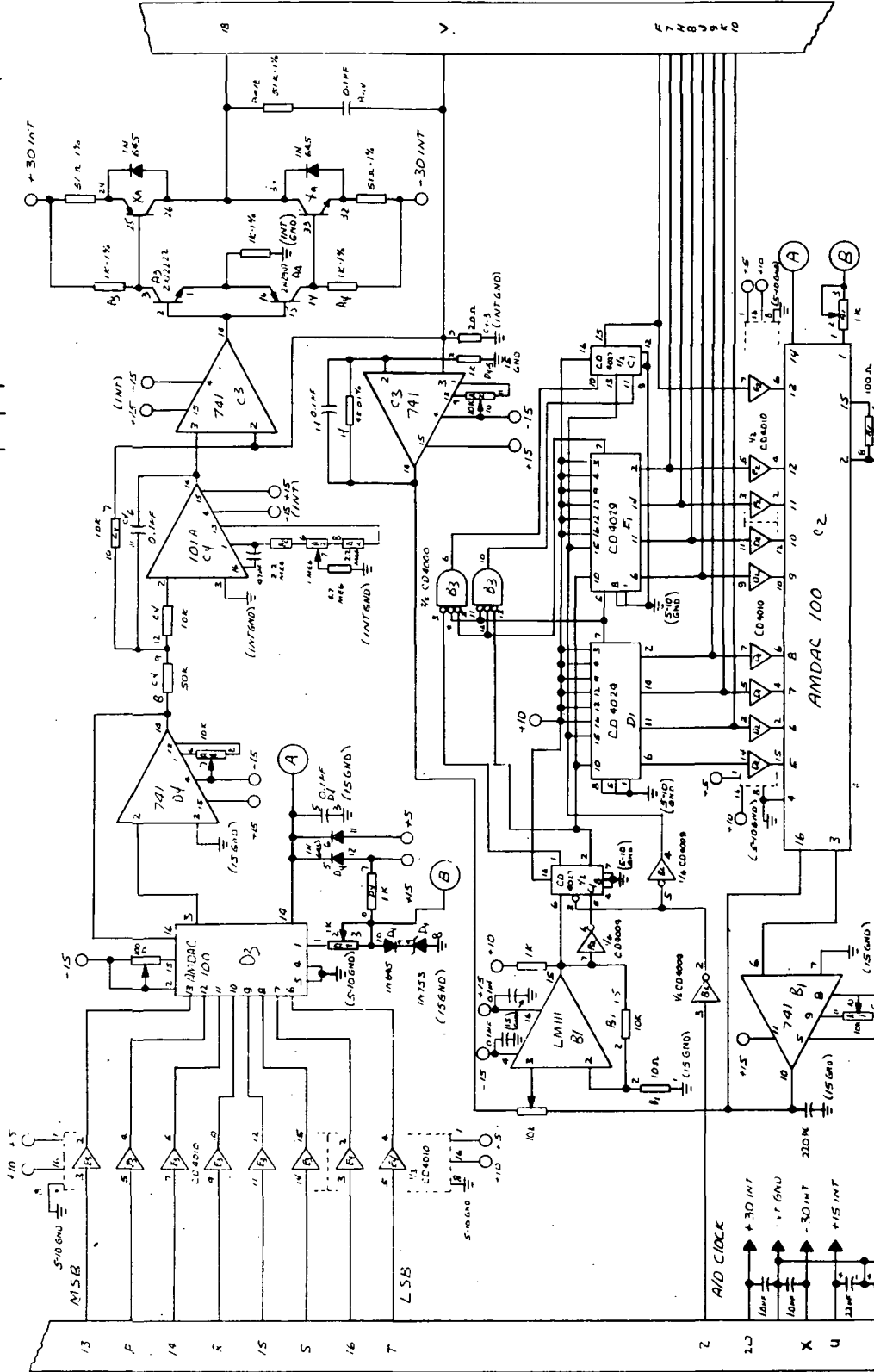
3.1.4 Data Serial & Storage Registers/Module No. 105; Location A1

The data recovered from board number 148 is fed to this module for loading and storage. Only the applicable 8 bits of the data word length are stored. The data inputs on pin 8 and is buffered by gates E1. The diode inputs on pins 7 and 9 are for Test Sequencer inputs used in Alternate "B" only. See Figure 16.

3.1.5 Servo Amplifier; A/D and D/A Converters/Module No. 29; Locations A3, A6, and A9

Figure 17 details the hardware for this module. Devices D3 and D4 make up the D/A converter which accepts the input data and converts it to the analog equivalent which is fed to the servo amplifier. The servo amplifier is comprised of devices C4 and C3 and the discrete transistors employed for current gain. This is a straight forward design which is a modification of an original MSFC design. A detailed discussion and applicable data is included in the appendix of this report.

A non-inverting amplifier with a gain of five amplifies the signal at the servo feed-back resistor which is located at C3. This analog signal is then digitized to 8 bits by the A/D converter detailed in the lower half of the schematic. The word derived is out-putted to an 8 bit comparator for comparison against the input digital word. As can be seen, common reference voltages are derived for use by the D/A and A/D converters. The A/D converter is comprised of a comparator, (LM111; B1) an up/down counter (CD4029's at D1 and E1) and a D/A converter (AIMDAC1000 at C2). To

[illegible]

track the incoming analog signal, the counter is made to count in the direction that will generate a D/A voltage equal to the incoming voltage. Once that point has been reached, the A/D will alternate between the up count and the down count which will alternate the ninth least-significant-bit up and down in value. Of these nine bits, only eight are used for the conversion output. This technique insures stability of the output data.

3.1.6 Eight-Bit Comparator ± 2 Bits/Module No. 58; Locations A2, A5 & A8

The purpose of this module is to ascertain whether or not the servo amplifier and the accompanying A/D and D/A converters are performing within the 2-bit tolerance. The 8-Bit Comparator, illustrated in Figure 18, compares the input data word with the digitized servo voltage and outputs a zero for $A < B$ and $A > B$ and a one for $A=B$ as long as the two words are within ± 2 bits. Should they differ by more than two bits at any time, the comparator issues a zero for A not equal to B and a one for A different from B . In the logic, the three least significant bits are examined for a numerical difference of two or three and a comparison of the third LSB. Where the difference between the numbers is three and the third LSB's agree, a 1 is added to the third LSB of the larger of the two inputs. Where the difference between the inputs is two and the third LSB's agree, no alterations are made to either number. Where the difference between the two inputs is two and the third LSB's disagree, a 1 is added to the smaller of the two inputs to make the third LSB's agree going into the 54L85 comparator.

When the difference between the two LSB pairs is one and the third LSB's disagree while the second LSB's agree and are both zero, no alterations are made at the input of the 54 L85. However, when the second LSB's are both

one, a one is added to both of the third LSB's. The resultant answer in both of these situations is identical since both number sets differed by three.

The advantage of this circuit is that it is static and it does not require any clocked operations such as would be needed for an add and shift comparison scheme.

3.1.7 Power Interrupt Logic/Module No. 60; Locations A4, A7 & A10

This logic, shown in Figure 19, controls the power to the servo amplifier. If the servo amplifier faults or if any of the mechanisms responsible for monitoring the failures fault, the servo amplifier is disabled and isolated by disconnecting the ± 30 volts and the ± 15 volts.

The automatic disable is retrieved from the 8-bit Comparator and inputs to pins 6, 7 and 8. Right at the transient state of detecting an error greater than two bits, an inhibiting one-shot is triggered such that the power is inhibited from reconnecting until the servo amplifier is given opportunity to settle. This type of situation arises when a step-function command suddenly appears at the data register and the servo amplifier has not yet responded. Latch E2 memorizes the input commands whether they are automatic as from the comparator or computer management routine or if they are manual as issued through the data terminal or test set via the supervisory command set.

The second half of the one-shot located at D2 provides an inhibiting function to the turn-off mechanism when the power interrupt is reset. Since the servo amplifier is off prior to the reset command, an immediate discomplate would

be experienced from the 8-bit comparator and this turn-off command is inhibited until the servo can, again, settle down. Four transistors are employed as switches to interrupt the power lines and the state of the latch E2 determines the on/off condition of the transistors.

Status is monitored by a divider on the 30 volt interrupted line and is buffered and sent to both the math model processor and the A.I. Status Monitor.

3.1.8 Computer Input Logic/Module No. 74; Location C3

This logic set is addressable by the computer and it multiplexes in the selected data. There are three data words that are selected which are input word, actuator position and delta pressure. The status word is comprised of the three servo amplifiers on/off status, cycle test which is the indicator that the computer is running, parity test of the scratch pad and control memories, and power fail which is the indicator for a transient power failure in the processor. See Figure 20 .

The circuitry interfaces the CMOS inputs to the TTL level and the selected inputs are placed directly on the processor's data memory bus. The processor's first 16 locations of data memory are dedicated to I/O.

3.1.9 Computer Output Logic/Module No. 75; Location C2

This module decodes the output addresses issued by the computer and stores the words that are on the data memory bus at the time. It also decodes the addresses employed for the input logic and outputs a set of enable lines to the

Computer Input Logic module previously discussed. Figure 21 shows this design.

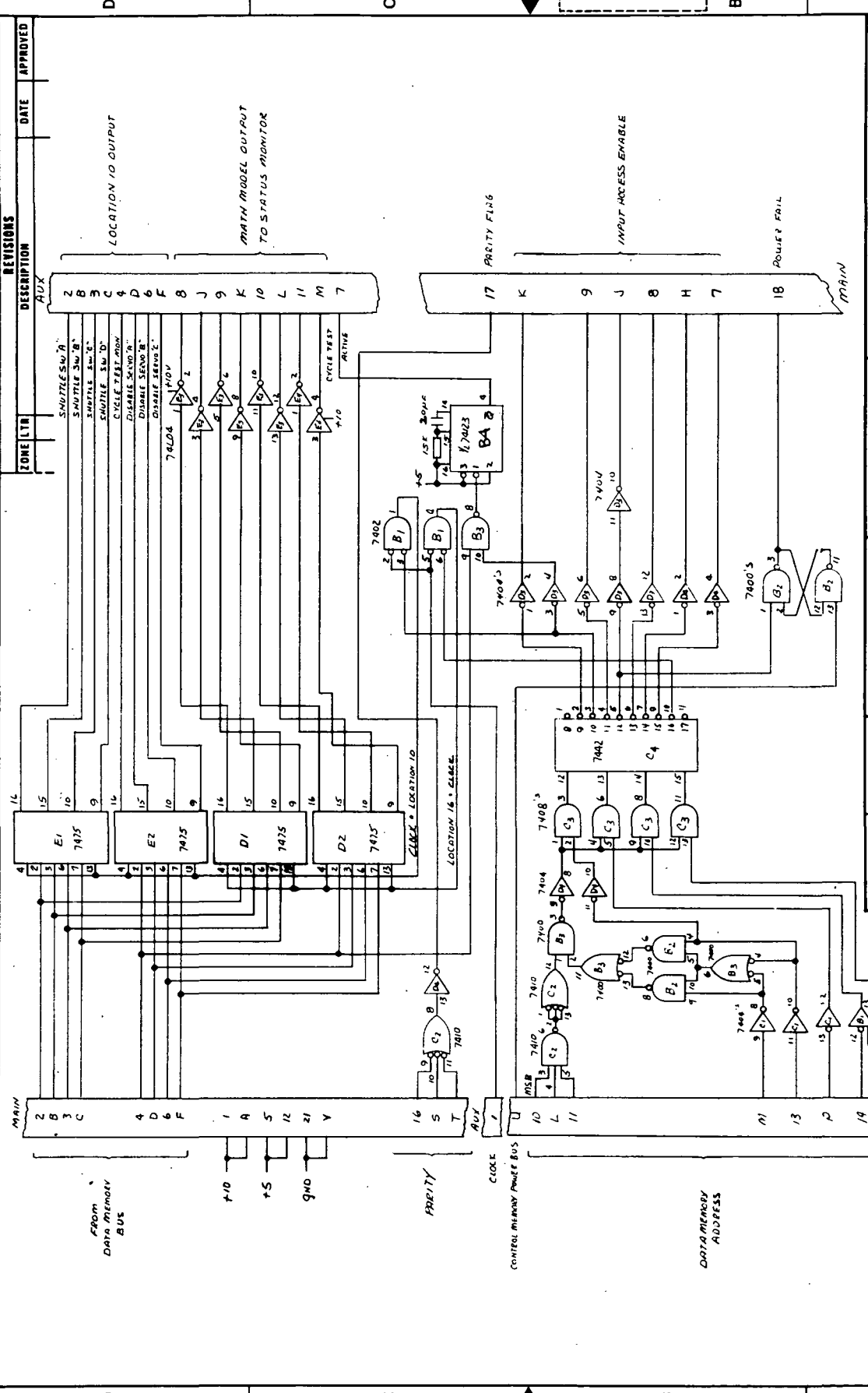
The stored output words contain the calculated actuator position, commands to shuttle switches "A", "B", "C", and "D" disable commands for servo amplifiers "A", "B", and "C", and a monitor for the cycle test flag. The information is stored in devices E1, E2, D1 and D2. The logic feeding C4 reduces the data memory address to four lines for decoding of the enable commands to read or load the I/O locations. The one-shot at B4 senses the cyclic accessing of location 3 and issues a continuous re-triggered low to indicate that the processor is running. Should the processor stop its 4ms iterations, the cycle test will indicate this failure such that the previous calculations are deemed invalid.

The latch at B2 memorizes any power transients that may have occurred which could also have upset the calculation process. The power fail flag also is an indicator that the previous calculations are invalid. If Alternate "A" had been assembled in its original redundant configuration, the cycle test, parity and power fail flags would have been used as part of the diagnostic scheme to verify that each of the three processors were, in fact, performing normally. The present use, however, is to reset and re-calculate within the simplex processor.

3.1.10 Computer Clock Generator/Module No. 69; Location B11

Figure 22 illustrates the hardware required to generate a 2Mhz continuous clock from the 1 Mhz systems clock that has every 20th pulse missing.

The N562 at A2 is a phase locked oscillator that operates at or near 4 Mhz in its free-run mode. It is phase locked at 4 Mhz to the 1Mhz input clock by dividing down the output to 1 Mhz and comparing it to the input clock. The 2 Mhz clock is derived from the divider and will always be phase related to the incoming clock.



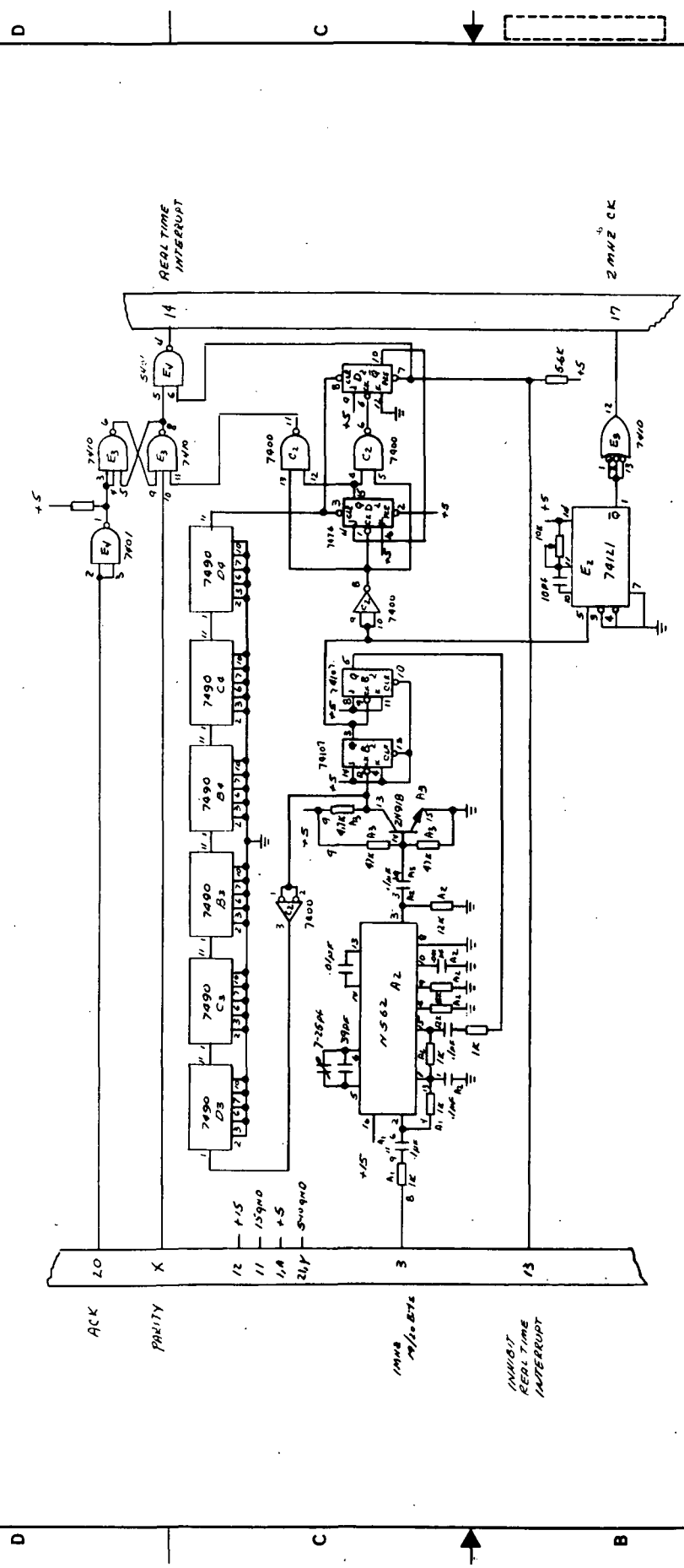
ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A DIVISION OF NCR		CONTINUED	
ACTUATOR INTERFACE ALI-A		UNLESS OTHERWISE SPECIFIED	
MODULE NO. 75		DIMENSIONS ARE IN INCHES	
COMPUTER OUTPUT LOGIC		TOLERANCES: XX ± 0.2, XXX ± 0.10	
SIZE CODE IDENT NO. C 00724		MATERIAL	
SCALE		FINISH	
FIG 21		APPLICATION	
		NEXT ASSY	
		USED ON	
		APPROVED	
		MAINT ENGR	
		APPROVED	
		SHEET	

REVISIONS	DESCRIPTION	DATE	APPROVED
1	ZONE 1/2		

2

3

4



ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF MCG	
ACTUATOR INTERFACE ALT. "A"	
BOARD NO 69	
COMPUTER CLOCK GENERATOR	
SIZE	CODE IDENT NO
C 00724	FIG 22
APPROVED	MAINT ENG
APPROVED	APPROVED
REPT ASSY	USED ON
APPLICATION	
FINISH	

This feature eliminates any time correlation problems between the input word and computer timing.

A set of 7490 decade dividers is used to develop the real-time interrupt.

The real-time interrupt insures that the processor calculates the actuator position on a real time basis and that it tracks precisely the position movements of the actuator. The real time interrupt occurs every 4 milliseconds and sets latch E3. The latch resets by an acknowledge command issued by the processor which indicates that the interrupt has been accepted and recognized.

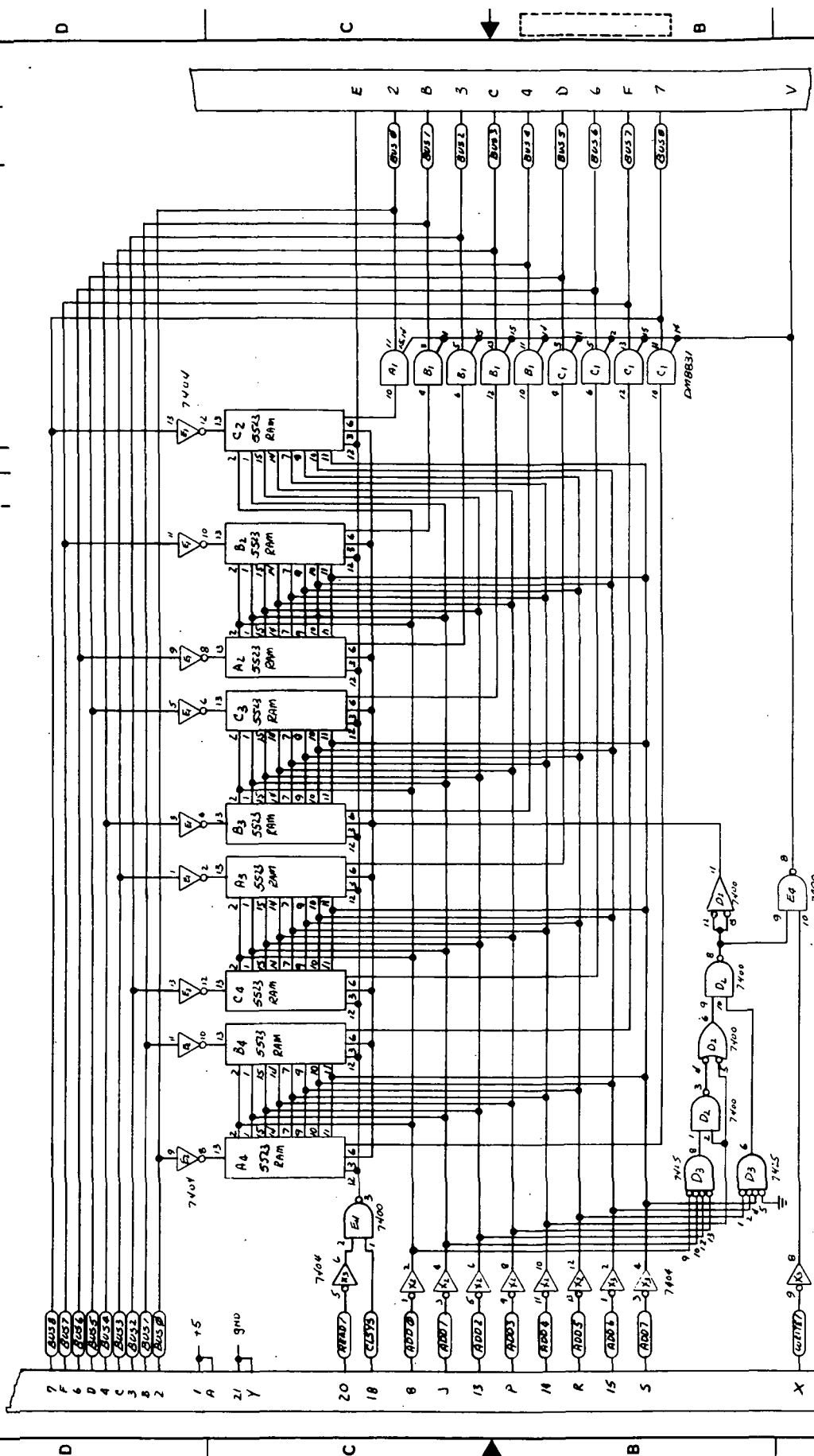
An inhibit function is provided to stop the real-time interrupt from occurring so that the processor program may be examined on a single-step basis from the maintenance panel.

3.1.11 Scratch Pad Memory/Module No.81; Location C4

This memory is 64 words by 9 bits where each word has 8 bits of data plus one bit for parity. The input and output of the memory is a single bus that is driven with a set of tri-state line drivers. Therefore, information is either taken off of the bus or it is put on to the bus. The terminology is bus oriented also which means that the "write" function writes data on to the bus from the memory and the "read" function reads data from the bus. This is not to be confused with the read/write functions of the individual memory devices where reading extracts the data in memory and writing places data in memory.

The address bits are inputted on the left of Figure 23 as is the read and write controls. The address is examined for location number and the first 17 (0-16) locations are inhibited since they are allocated for I/O. The CLSYS input is the systems clock.

REVISIONS		
ZONE	DESCRIPTION	DATE



		NEXT ASSY APPLICATION	USED ON	FINISH	MATERIAL	BREAK SHARP EDGES .015 ± .010 REMOVE BURRS G17 REL	APPROVED	MAINT ENGR	SIZE	CODE IDENT NO	C 00724	F16 23	SCALE	SHEET		ELECTRONIC COMMUNICATIONS, INC. ST PETERSBURG, FLORIDA • A SUBSIDIARY OF NCR	
																	UNLESS OTHERWISE SPECIFIED
																	DIMENSIONS ARE IN INCHES
																	.XX ± .02, .XXX ± .010
																	TOLERANCES: ANGLES ±
																	HOLE DIA TOLERANCES
																	HP
																	THRU
																	500 DIA
																	500 DIA

3.1.12 Random Access Control Storage/Module No. 82; Locations C7 & C8

Random Access Control Storage/Module No. 83; Location C9

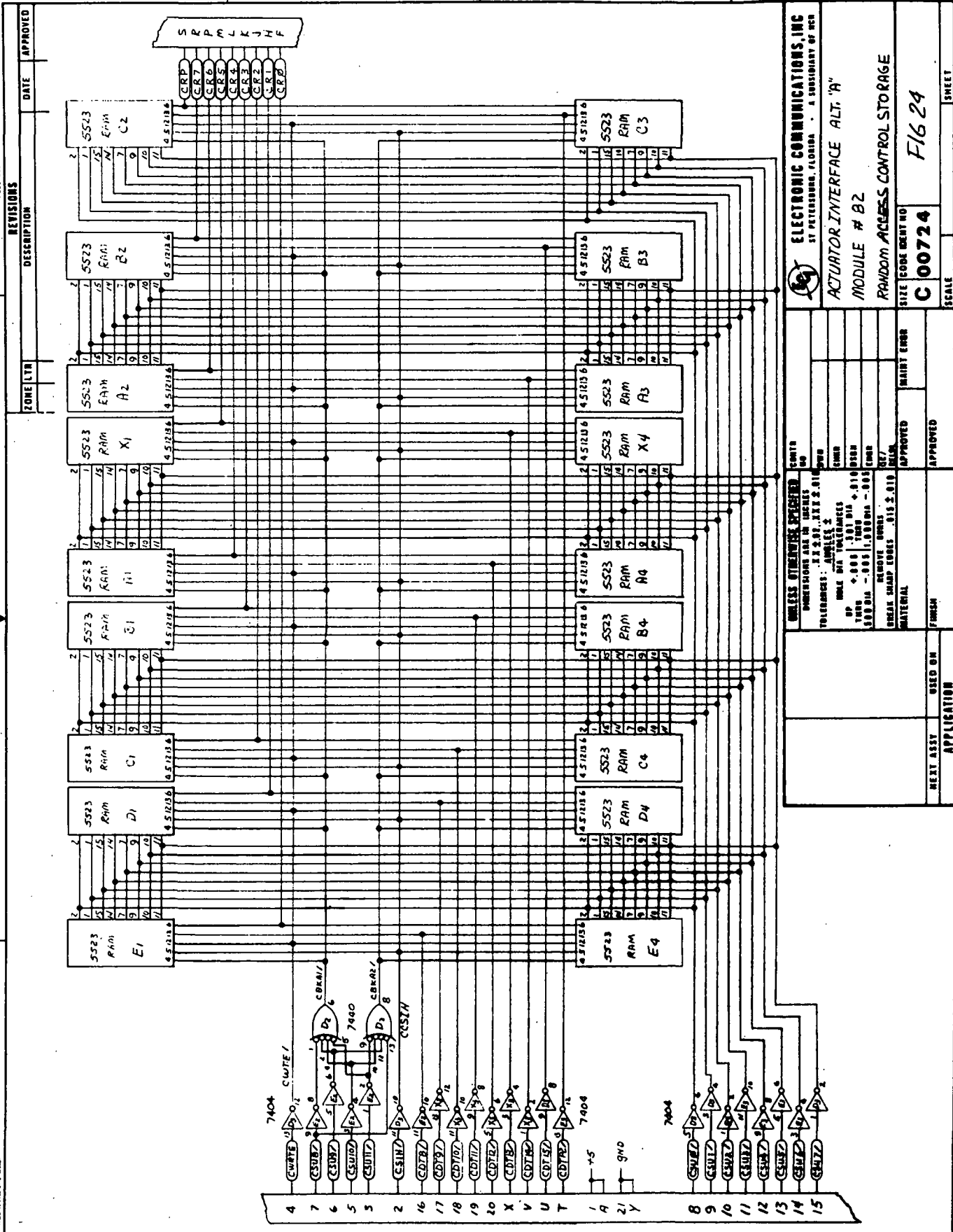
These modules comprise the complete control storage memory required for the math model processor. Random Access Memory is employed to maintain the flexibility of programming. However, normally, the control storage would be read-only storage.

Each of the modules of the No. 82 type will store 512 8-bit words and each is used to store CR and RAD information respectively. Module No. 83 will store 256 8-bit words of CR and 256 8-bit words of RAD yielding a complete total of 768 control storage words which are 16 bits in length. As can be seen, memory is allocated for parity for each 8-bit section and they are designated as CRP and RADP respectively. The systems block diagram illustrates the mutual interconnects of these three memory cards. Figures 24 and 25 are the schematic diagrams of Modules 82 and 83 respectively.

The memories are loaded with their instructions via the tape reader and its associated electronics. And, it has to be re-loaded after power has been interrupted.

3.1.13 Memory Access Multiplexer/Module No. 73; Location C10

The purpose of this module, illustrated in Figure 26, is to select the address controller of the control storage memory. That is, whether the processor or the tape reader will control the address. This control is a maintenance panel function. In the "Computer Run" mode, the processor increments the control storage and in the alternate modes, the tape reader electronics control the memory for loading.



ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF NCR	
ACTUATOR INTERFACE ALT. "A"	
MODULE # B2	
RANDOM ACCESS CONTROL STORAGE	
SIZE CODE IDENT NO C 00724	SHEET FIG 24
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES: .0025" MAX 2.00" MIN FINISH: .0005" MAX UP: .0005" MAX DOWN: .0005" MAX THRU: .0005" MAX 100 DIA: .0005" MAX BREAK SHARP EDGES: .015" MAX MATERIAL: 6061-T6 ALUMINUM REMOVE BURRS APPROVED: [Signature] MAINT ERROR: [Signature]	
NEXT ASSY USED ON APPLICATION	APPROVED: [Signature]

3.1.14 Parity Generator and Check Logic/Module No. 72: Location C6

This module checks the parity of the instruction set and of the data memory while it is in use with the processor. The data memory parity section both reads and writes parity onto the bus depending on the mode of the data memory.

Parity errors are flagged to the processor as a probably failure in the previous calculations, and a continuous parity error will cause shutdown of the processor. The schematic is illustrated in Figure 27.

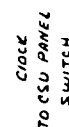
3.1.15 Tape Reader Interface Clock/Module No. 70; Location C12

This logic generates the timing and control for the tape reader and the verification of the extracted data as well as verification of the data written into memory.

Gate B1, device C1, divider A1 along with A2 comprises the basic clock and increment pulse generator for the tape reader. The speed is alterable between 100 Hz and 20 Hz for a slow and fast read speed. Providing that no error is detected in the data handling, the clock will be permitted to run as controlled by latch D3. The remainder of the logic controls the specific number of allowable tape increments whether it be for one character, a complete word or a continuous run of successive valid words. See Figure 28.

3.1.16 Tape Reader Interface; Data Acquisition and Verification Logic/Module No. 71; Location C11

The paper tape has eight perforable channels where a hole represents a one. The assignment for the eight channels are, from left to right, parity, start, stop or end, memory designate and four bits for information. Now, it takes eight increments

[illegible]

123456	345678	APPLICATION
123456	345678	345678

of the tape for a complete word; the first four increments are for data to be stored at that address.

For each increment of the tape, parity is assigned and checked by device B4, See Figure 29. When a start bit is sensed, the logic stores the data collected in the next four increments as a memory address and the following four increments as memory data. Next, an end bit has to be sensed as well as the increment counts being correct before the memory is written with the collected data or before the tape reader is permitted to continue. Once the memory is written, a read cycle is exercised and the returning data is compared with the source data to insure that the data has been entered correctly. Gate A3 Nor's the potential errors just discussed which are miscount, tape increment parity fail and memory read discompare. Devices D4, D3, D2, D1, E4, E3, E2 and E1 comprise the information shift register/accumulator which shifts the sequential information from the tape in four-bit blocks. At the end of a complete memory location/data word, devices B2 and B3 assign parity to the data 8-bit sections. After writing the memory, devices C4, C3, C2, C1 and B1 compare the memory read-out with the stored word in the shift register. Providing no error occurs, the tape reader is then permitted to increment on through the next character, word, or continuous function as controlled on the maintenance panel. Table 3-1 illustrates the tape format discussed.

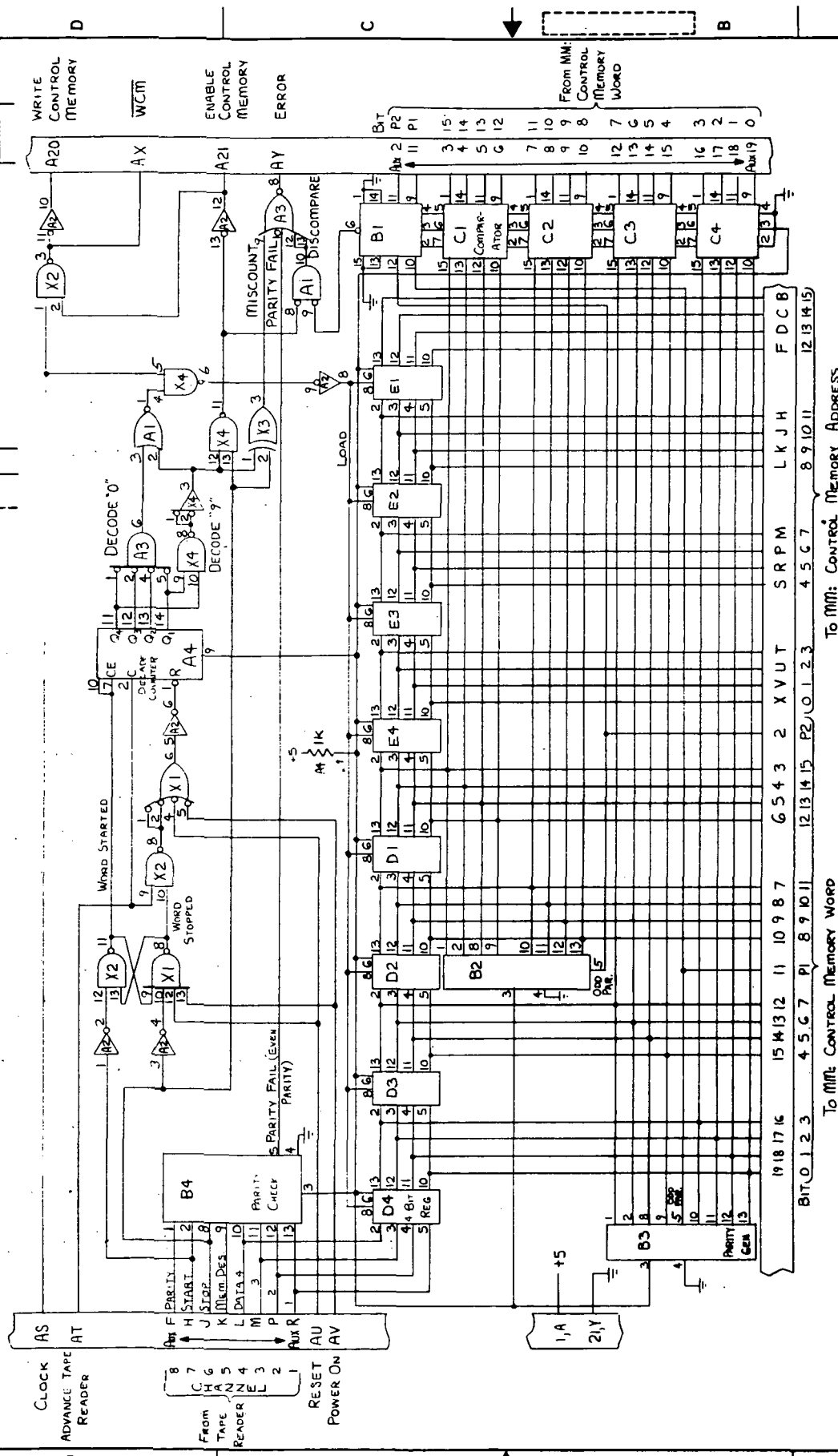
CHANNEL		8	7	6	5	4	3	2	1
START		x	1	0	x	x	x	x	x
ADDRESS	1	P	0	0	M	0	0	0	0
	2	P	0	0	M	0	0	0	0
	3	P	0	0	M	0	0	0	0
	4	P	0	0	M	0	0	0	0
DATA	1	P	0	0	M	0	0	0	0
	2	P	0	0	M	0	1	1	0
	3	P	0	0	M	1	0	1	0
	4	P	0	0	M	0	1	0	0
END		x	0	1	x	x	x	x	x

TABLE 3-1
PERFORATED TAPE FORMAT

REVISIONS	DATE	APPROVED
1		

ZONE/LTR	DESCRIPTION

DATE	FILE NO.	14



ELECTRONIC COMMUNICATIONS, INC. 517 PETERSBURG, FLORIDA		AIU ALT "A" BREADBOARD TAPE READER INTERFACE - DATA ACQUISITION AND VERIFICATION MODULE	
SIZE CODE IDENT NO C 00724		FIG 29	
SCALE		SHEET	

LOCATION	DEVICE	W/PINS	VCC	GRD
X1	SN7440	14	14	7
X2, X4	SN7400	14	14	7
X3	SN7486	14	14	7
A1	SN7402	14	14	7
A2	SN7404	14	14	7
A3	SN7425	14	14	7
A4	SN7410	14	14	7
B1, C1, C2, C3, C4	SN7485	16	16	8
B2, B3, B4	SN74180	14	14	7
D1, D2, D3, D4, E1, E2, E3, E4	SN7495	14	14	7

UNLESS OTHERWISE SPECIFIED		CONTS
DIMENSIONS ARE IN INCHES		NO
TOLERANCES: .005, .010, .015, .020, .030, .040, .050, .060, .070, .080, .090, .100, .120, .150, .200, .250, .300, .350, .400, .450, .500, .600, .700, .800, .900, 1.000		NO
HOLE DIA TOLERANCES		NO
TAPER: .0001, .0002, .0003, .0004, .0005, .0006, .0007, .0008, .0009, .0010, .0012, .0015, .0020, .0025, .0030, .0035, .0040, .0045, .0050, .0060, .0070, .0080, .0090, .0100, .0120, .0150, .0200, .0250, .0300, .0350, .0400, .0450, .0500, .0600, .0700, .0800, .0900, .1000		NO
BREAK SHARP CORNERS		NO
MATERIAL		NO
APPROVED		APPROVED
MAINT ENGR		APPROVED
FINISH		APPROVED
NEXT ASSY		APPROVED
USED ON		APPROVED
APPLICATION		APPROVED

Note: "M" was assigned as the "memory designator" which would steer the control either to Control Storage or Data Memory. The original intent was to store constants in the Data Memory as part of the control program. This requirement has been deleted and, therefore, "M" is not used.

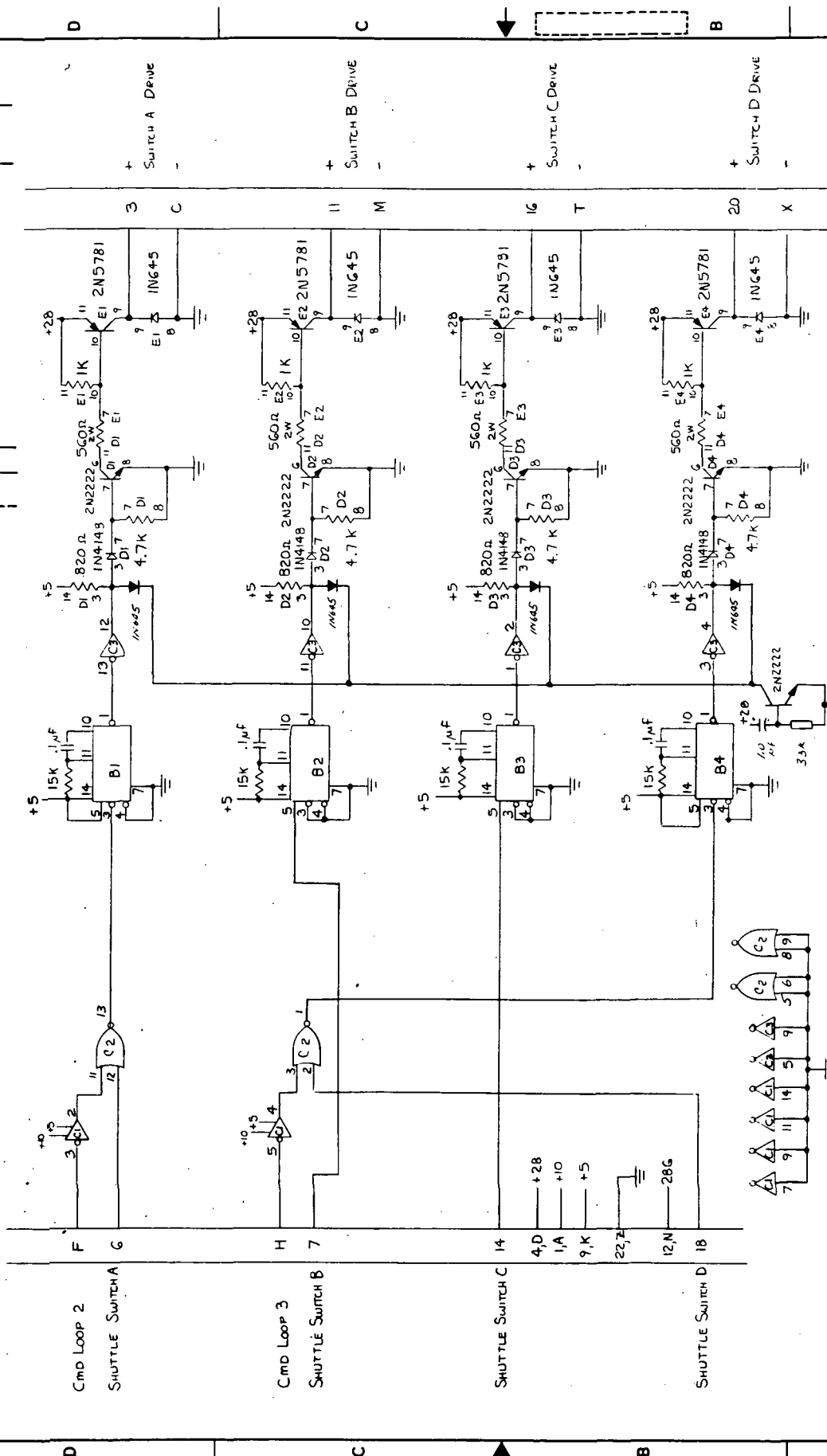
3.1.17 Hydraulic Switch Drivers/Module No. 62; Location C1

This logic, shown in Figure 30, acts upon commands recieved from the math model or the supervisory decoder. If a switch is commanded to shuttle, the command triggers a one-shot associated with the switch. Each one-shot issues a 1 millisecond pulse which is fed to a driver capable of driving the switch with 1 amp at 28VDC.

The 2N2222 and 2N 5781 comprise the 28VDC interface driver. A 2N222 transistor is employed for system initialization at power-on. Because there is a tendency of the one-shots to trigger or generally exhibit some instability as the power comes up to voltage, the 2N2222 common to all drivers acts as an inhibitor until the 28VDC has stabilized. This is accomplished by charging a capacitor through the base of the transistor which will hold the transistor on until the 28VDC has nearly reached its maximum value. As the charging current approaches zero, the transistor will turn off and the individual current drivers will then be permitted to be controlled by the logic one-shots. As can be seen in the diagram, the stages are seperated and isolated by the diodes which clamp the inputs of the drivers during turn-on.

Each of the switches may be commanded individually by the supervisory input or an operational loop may be commanded by the computer. A computer "reset" will erase the previous computer command and the test set reset will un-latch the switch and return it to its normal state.

REVISIONS	DATE	APPROVED
1		



ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A DIVISION OF RCA	
AIU ALT. "A" BREADBOARD HYDRAULIC SWITCH DRIVERS (6.2)	
SIZE (CODE IDENT NO)	C 00724
SCALE	FIG 30
UNLESS OTHERWISE SPECIFIED	CONTR NO
DIMENSIONS ARE IN INCHES	NO
TOLERANCES: .XX ± .02, .XX ± .010	OWN
ANGLES: .XX ± .02	OWN
HOLE DIA TOLERANCES	OWN
UP +.001, .501 DIA +.010	OWN
DOWN -.001, .501 DIA -.003	OWN
100 DIA ±.003, 1.000 DIA ±.003	OWN
BREAK SHARP EDGES: .015 ± .010	OWN
MATERIAL	APPROVED
FINISH	APPROVED
USED ON	APPROVED
APPLICATION	APPROVED
SN54121	SN54121
SN54009	SN54009
SN5402	SN5402
SN5405	SN5405

3.1.18 Dual Tracking A/D Converter/Module No. 20; Location B10

The purpose of these A/D Converters is to digitize both the actuator position voltage and the pressure sensor voltage. The design of the A/D converter is identical to the one employed on the servo amplifier board as a monitor.

Figure 31 again illustrates the hardware except that a dual coordinate system is assigned for each device. As in the previously discussed converter, the 8 most significant bits are outputted. Bit 9 constantly alternates when the converter reaches equilibrium and bit 10 of the D/A portion is not used.

3.1.19 Status Monitor Timing and Control/Module No. 61; Location B4

The function of this logic shown in Figure 32 is to establish the proper timing for retrieving information requested by a supervisory command and generating the send window 40 microseconds after the request event.

The supervisory commands which are to be answered, input to this logic at the upper left of the schematic and are buffered and sent to the data multiplexer selector. Devices D1 and D2 detect that only an add number of requests occurred at any given time. That is, this network will set latch C2 only if one command was sensed and not if two commands occurred simultaneously. (The probability of three simultaneous commands is remote so actually the logic is considering the difference between one and two.)

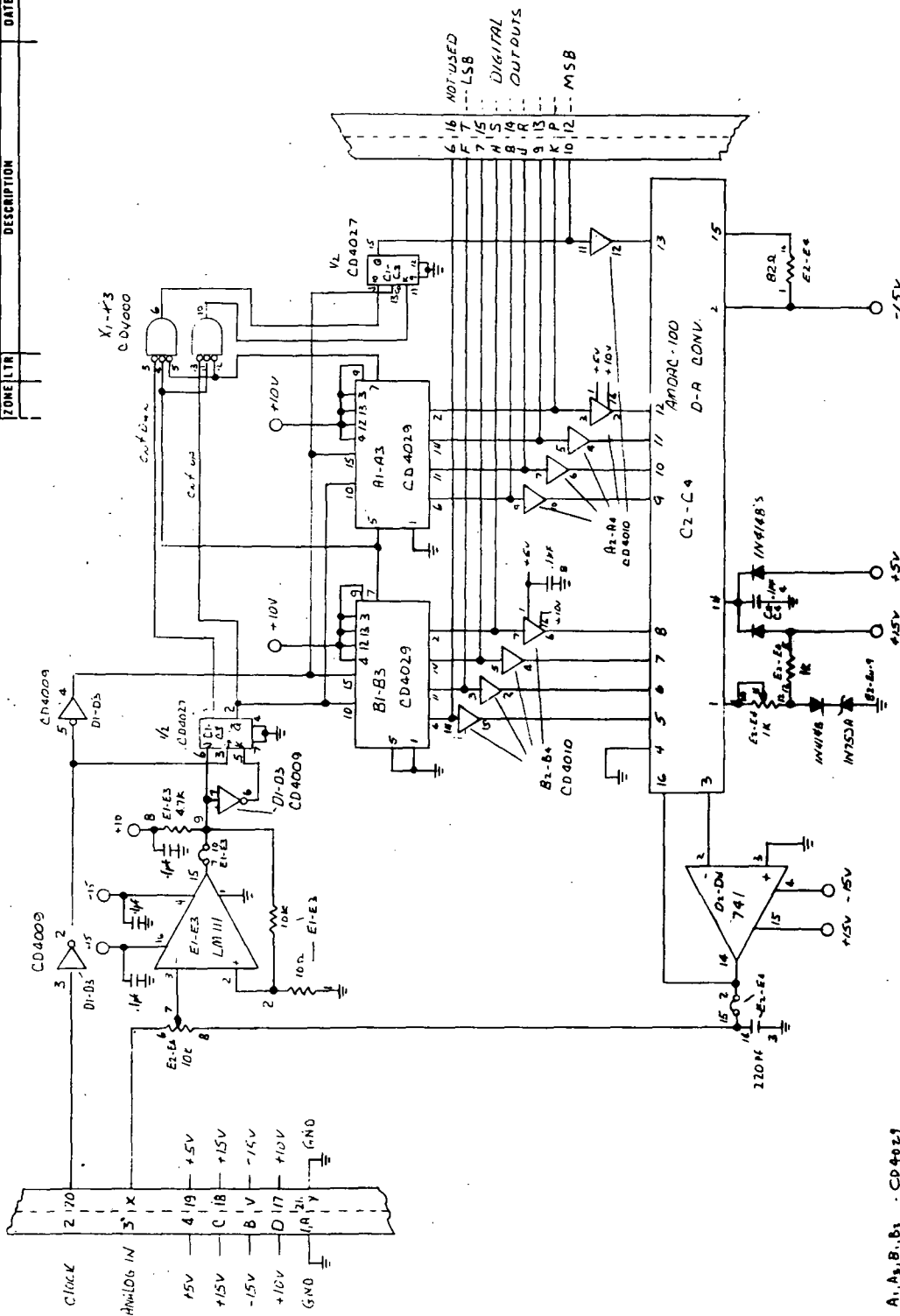
Counter C3 counts the end-of-word pulses and opens a 20 usec gate at the end of 40 usec after the receipt of command. This gate, D3, formulates the Status Monitor shift clock. The Polar Rz Modulator clock is derived from the status monitor shift clock but is delayed sufficiently until the data has successfully propagated in the status monitor shift register. Also, each pulse of the clock is stretched to 500 ns by the network at E4 so that the data has the proper format out of the Polar Rz driver.

REVISIONS	DATE	APPROVED
1		

2

3

4



COMPANY PROPRIETARY-PRIVATE HANDLING

ELECTRONIC COMMUNICATIONS, INC. 31 PETERBORO, FLORIDA - A SUBSIDIARY OF MCR	
ATU BREADBOARD-ALT. "A" BOARD NO. 20 DUAL-TACKING A/D CONVERTER	
SIZE CODE MENT NO C 00724	FIG 51
SCALE SHEET	
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES: .XX ±.01, .XXX ±.010 FINISH: MILL SURF: 100% UP: 100% DOWN: 100% 100% DIA: .005, .005 DIA: .005 BREAK SHARP CORNERS: .015 ±.010 MATERIAL: ALUMINUM MAINT: EMBR APPROVED: [Signature] APPROVED: [Signature]	
MENT ASSY APPLICATION	USED ON APPLICATION

- A₁, A₂, B₁, B₃ - CD4029
 X₁, X₃ - CD4000
 D₁, D₃ - CD4009
 A₂, B₂, M₁, M₄ - CD4010
 C₁, C₃ - CD4027
 D₂, D₄ - LM741
 E₁, E₃ - LM111
 C₂, C₄ - AMDAC-100

3.1.20 Status Monitor Data Multiplexer/Module No.17; Location B6 & B8

The multiplexer selects any one of eight status words where each word is eight bits wide. The word select control is the supervisory command request and is labeled as the "WS" inputs to the schematic of Figure 33. It can be seen that this design is a straight forward multiplexer design with the word selected output to the lower left of the schematic.

The system employs two of these multiplexers so that the status word returned contains 16 bits of information. The two halves of the word are converged in Module No. 114.

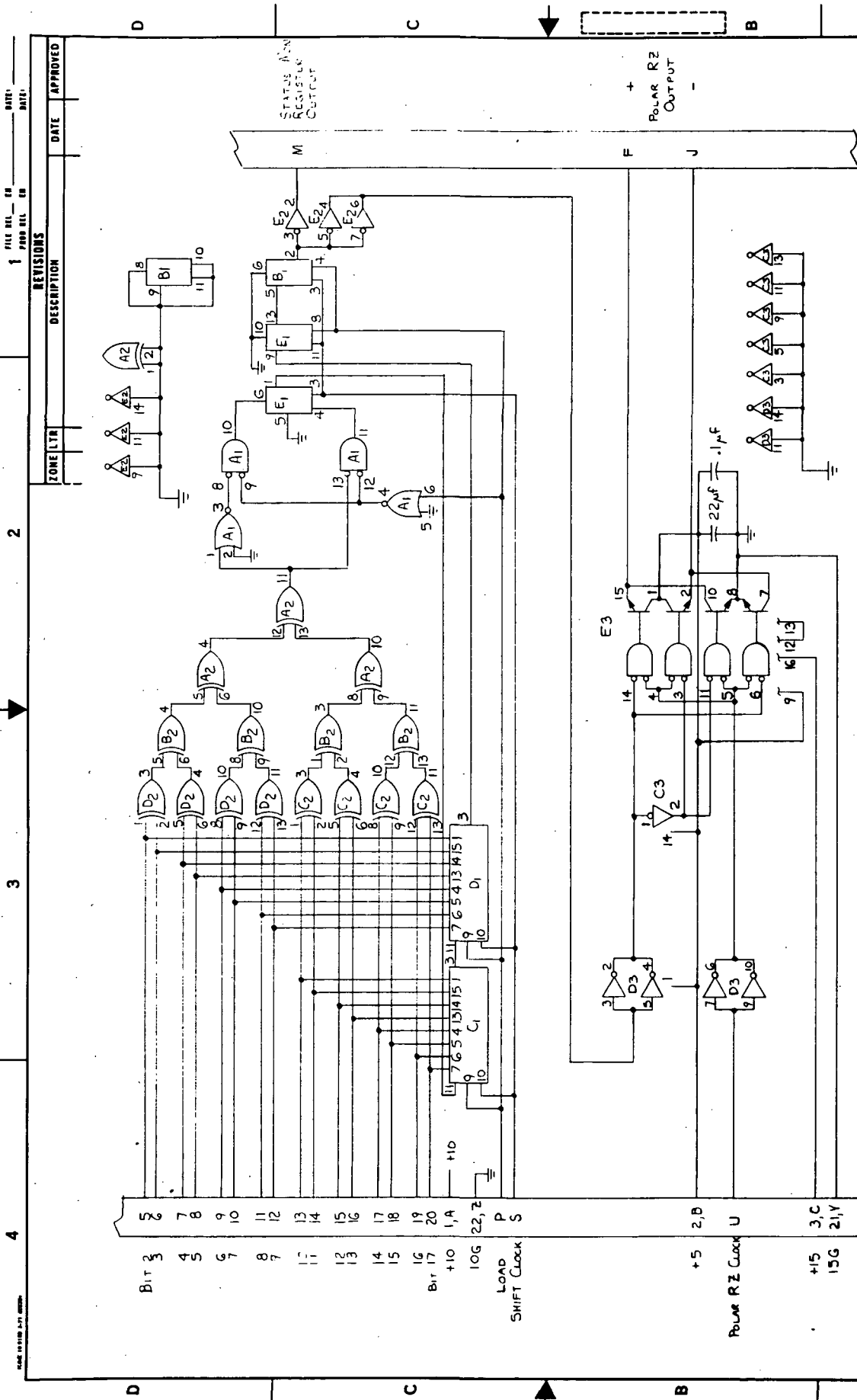
3.1.21 Status Monitor Parity Generator; Shift Register; Polar RZ Modulator/ Module No. 114; Location B7.

This logic serializes the multiplexed status word, adds parity and shifts it out to the data terminal or test set through the Polar RZ Modulator. See Figure 34.

Devices C1 and D1 are 8-bit shift registers in which the data is loaded. Devices E1 and B1 provide the shift space for the first two fixed bits and the 19th bit which is parity. The parity is derived from the parity tree formulated by exclusive Or's D2, C2, B2 and A2. The shift register output is buffered and sent to device E3 which is a SN55325 core memory driver employed as a Polar RZ modulator. The load pulse and clock are derived in the status monitor timing and control logic as is the polar RZ clock which inputs to this board.

3.1.22 Alternate "A" Power Supply/Module No. 64/Location B13

This power supply is a single switching inverter supply which provides the majority of the power to the Actuator Interface Alternate "A" unit. The processor power



ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF RCA		FIG 34	
AIU ALT. "A" - STATUS MONITOR PARITY GENERATOR, SHIFT REG., AND POLAR RZ MOD. MODULE 114		C 00724	
SIZE CODE BENT NO.		MAINT. ENGR.	
APPROVED		APPROVED	
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES: .XX ± .02, .XX ± .010 ANGLES: 2 HOLE DIA. TOLERANCES .000 ± .001, .001 DIA. ± .010 .010 DIA. ± .001, .010 DIA. ± .005 BREAK SHARP EDGES .015 ± .010 MATERIAL		FINISH	
NEXT ASSY		USED ON	
APPLICATION		SHEET	

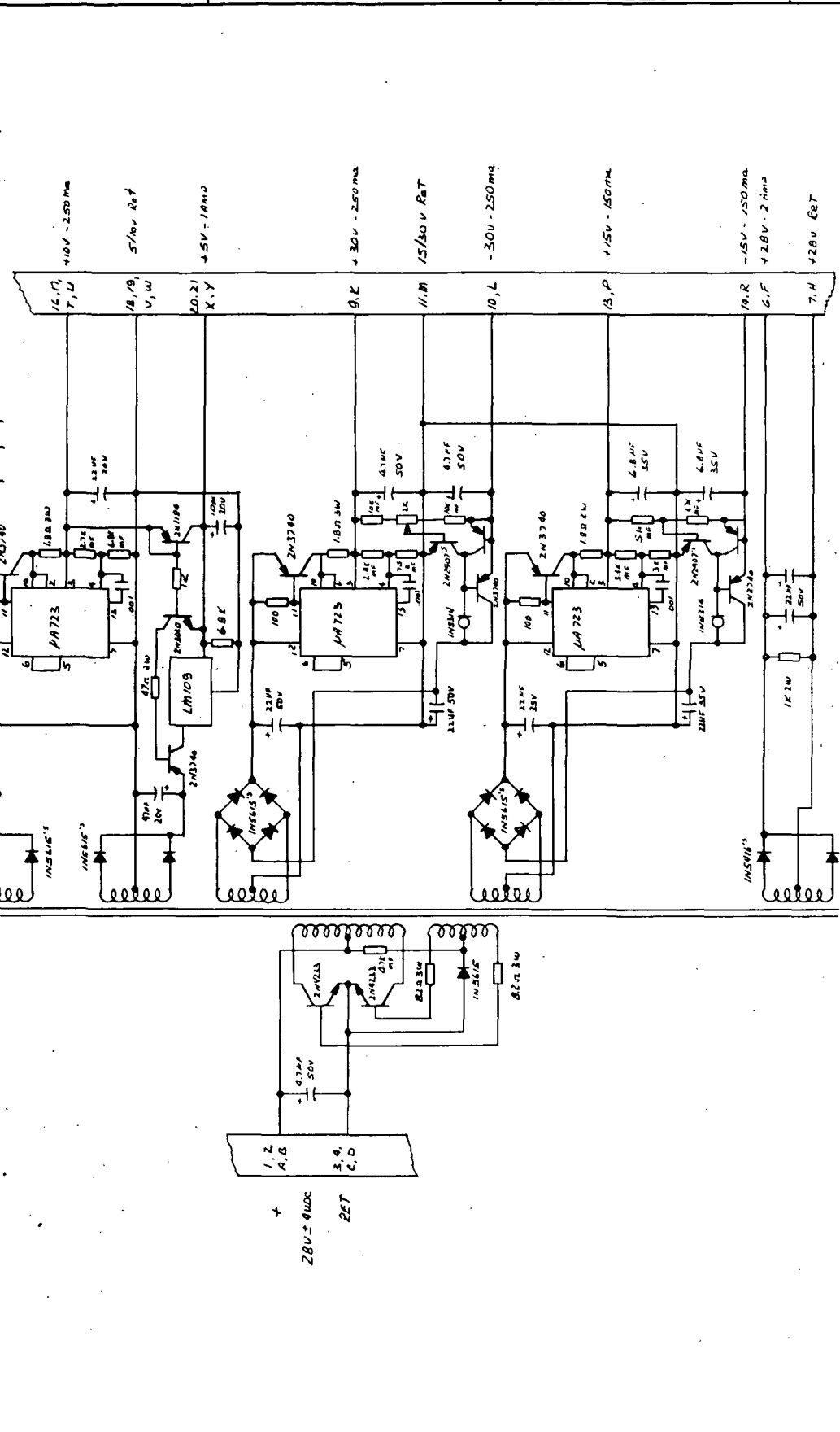
- A1 CD 4001
 A2, B2, C2, D2 CD 4030
 B1, E1 CD 4013
 C1, D1 CD 4021
 SN 5404
 CD 4009
 D3, E2 CD 4009
 E3 SN 55325

is provided by a separate supply. This supply utilizes i. c. regulators in five separate sections, each with its own transformer winding. The power supplied is:

- 28VDC @2A
- 15VDC @ 150 ma
- 15VDC @ 150 ma
- 30VDC @ 250 ma
- 30VDC @ 250 ma
- 5VDC @ 1A
- 10VDC @ 250 ma

The 5VDC and 10VDC have a common return; the ± 15 VDC and ± 30 VDC have a common return and the 28VDC has a separate return. See Figure 35. An important consideration in this supply is that the 10VDC always leads the 5VDC during the "on" time and that the 10VDC trails the 5VDC during the "off" time. This is necessary because of the characteristics of the CMOS CD4009 and CD4010. The output stages of these buffer/interface devices must always be at least 0.5VDC less than the input stage supply level. If not, the device can get into an avalanche mode not unlike a four-layer device where the current conduction becomes very high. This can destroy the interter plus overload the associated supply. Forcing these supplies to track prevents this possibility. However, it should be noted that because this system is fabricated in breadboard fashion, it can be easy to short pins during probe testing. If the 10 VDC was inadvertently shorted to ground for a short period of time, it is likely that the 5VDC did not track this momentary short. Some of the CD4009's and CD4010's can avalanche and will hold this mode until the power is cycled off and on. Should the system suddenly cease to operate normally during probing, this is likely the problem and it can be restored by cycling the power. Certainly, it can be seen that within a normal architecture this would not be a problem.

REV	DESCRIPTION	DATE	APPROVED
1	1000	10/1/68	



ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF RCA	
Actuator Interface ALT #1	
Module No 64	
ALT #1 Power Supply	
SIZE	CODE IDENT NO
C 00724	
SCALE	SHEET
SCALE	Fig 35

UNLESS OTHERWISE SPECIFIED	CONTR
DIMENSIONS ARE IN INCHES	NO
TOLERANCES: .XX ± .02, .XX ± .01	OWN
ANGLES: .XX ± .02, .XX ± .01	CHN
HOLE DIA TOLERANCES	CHN
BP	301 DIA ± .010 DISM
THRU	300 DIA ± .005 THRU DIA ± .005 ENGR
REMOVE BURRS	DE/
BREAK SHARP EDGES .015 ± .010	REL
MATERIAL	
APPROVED	MAINT ENGR
APPROVED	
FINISH	
USED ON	
APPLICATION	

3.2 Alternate "B" Circuit Modules

3.2.1 Line Receiver/Module No. 48; Locations A3 & B3

The schematic of Figure **36** details the line receiver employed in both the Input Data Validity Comparison (IVDC) and Input Supervisory Validity Comparison (SVC) functions. The schematic shows the hardware used for two out of the four lines inputting to each section. Each component has a dual coordinate system which then accounts for the four channels.

Each receiver is like the receiver described for Alternate "A" in that the lines are terminated in 75 ohms and the data is coupled and isolated by pulse transformers into a transistor data recovery circuit. However, for each channel, both the ONE's and ZERO's information is stripped off the incoming Polar RZ waveform for separate processing. It can be seen that each data recovery transistor can be inhibited by a parallel transistor. This is a test function to verify the performance of the line receiver section by section.

3.2.2 Serial Voter/Module No. 4; Locations A5, A6, A8, B5, B6, C5 & C6

Figure **37** illustrates the four-line-in, four-line-out serial voter. The function of the voter is to output four lines of like data that is immune to any two like or unlike failures on the input lines. For instance, two input lines could be shorted to ground, or a particular voltage, or one shorted to ground and another shorted to a voltage, or one shorted to ground and another could carry an unrelated signal; all of these failures would be discounted by the serial voter and the output lines would contain the valid data that has to exist on at least two input lines.

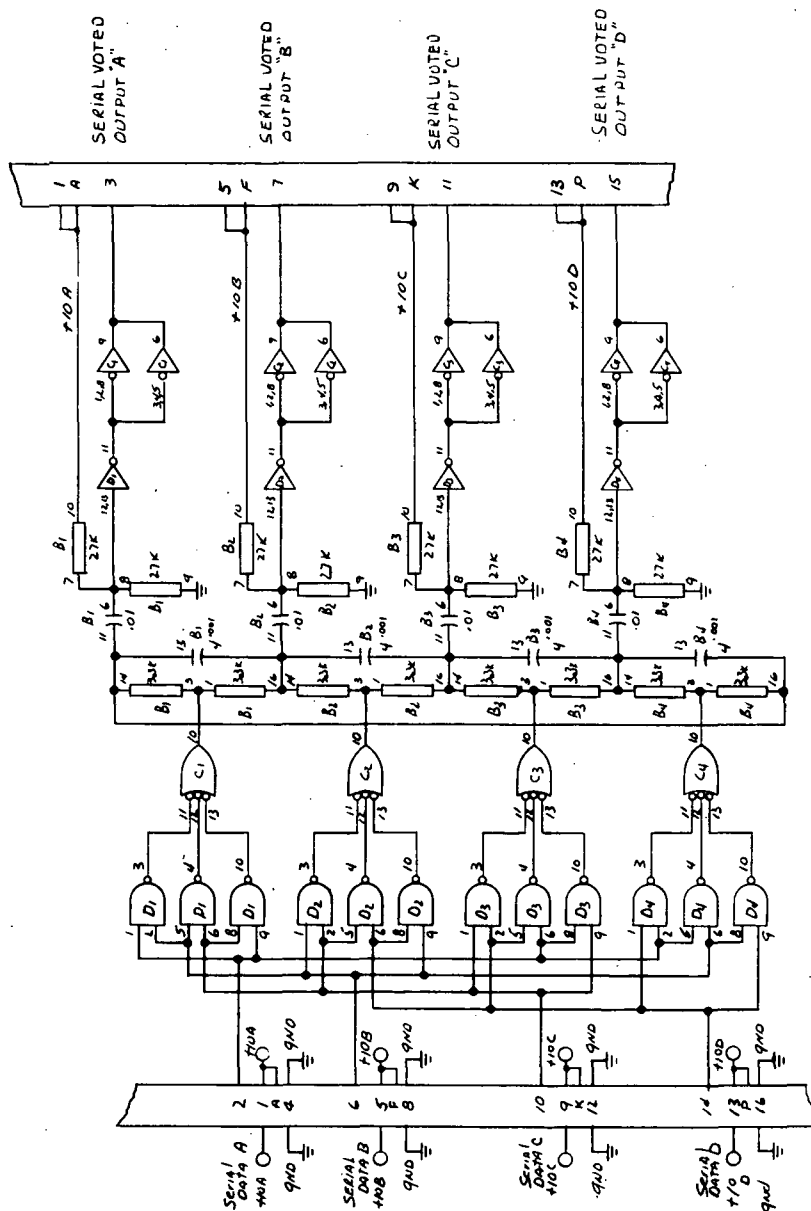
FILE REL. EN DATE: 1

2

3

4

ZONE	LTR	DESCRIPTION	DATE	APPROVED



COMPANY PROPRIETARY-PRIVATE HANDLING

D1-D4 = CD 4011 4EA
 C1-C4 = CD 4013 4EA
 B1-B4 = 33K RES
 A1-A4 = 27K RES
 4EA .001 CAP
 4EA .01 CAP

ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG DIVISION ST. PETERSBURG, FLORIDA		ALU BREADBOARD - ALT. "B" BOARD NO. 4 SERIAL VOTER	
CONTR. NO.	DATE	SIZE	CODE IDENT NO.
DRAWN	CHECKED	MAINT. ENGR.	APPROVED
DESIGNED	PROJ. ENGR.	RELIABILITY/QUAL. CONT.	APPROVED
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ON:		FINISH	
FRAC.	DECIMALS	ANGLES	
±.005	±.005	±.005	
REMOVE BURRS & BREAK SHARP EDGES		MATERIAL	
NEXT ASSY		USED ON	
APPLICATION		SCALE	
C 00724		F16 37	
SHEET		SHEET	

The input lines feed into a set of four majority voters where the outputs of these voters vote on similar signals two-out-of-three in four different combinations. The voters outputs then feed a resistive/capacitive network where the voting is uniquely combined to eliminate DC failures or interfering-signal failures. The bias networks at the input of the buffer inverters permit recovery of the attenuated signal levels which, in turn, feed buffer drivers.

3.2.3 Systems Clock Generator/Module No. 19; Location B7

The purpose of this module is to derive the system clock from the input supervisory serial words. It is fed from the serial voters with four separate inputs/pairs of voted "ones" and "zeros." The 1's and 0's are Ored at the input and then logically differentiated to generate a clock pulse of approximately 200ns. Also, from this Oring circuit, a missing-pulse detector senses the 20th bit frame and generates an "end-of-word" pulse for loading the collected input information. The EOW is shaped by a logic differentiator. See Figure 38.

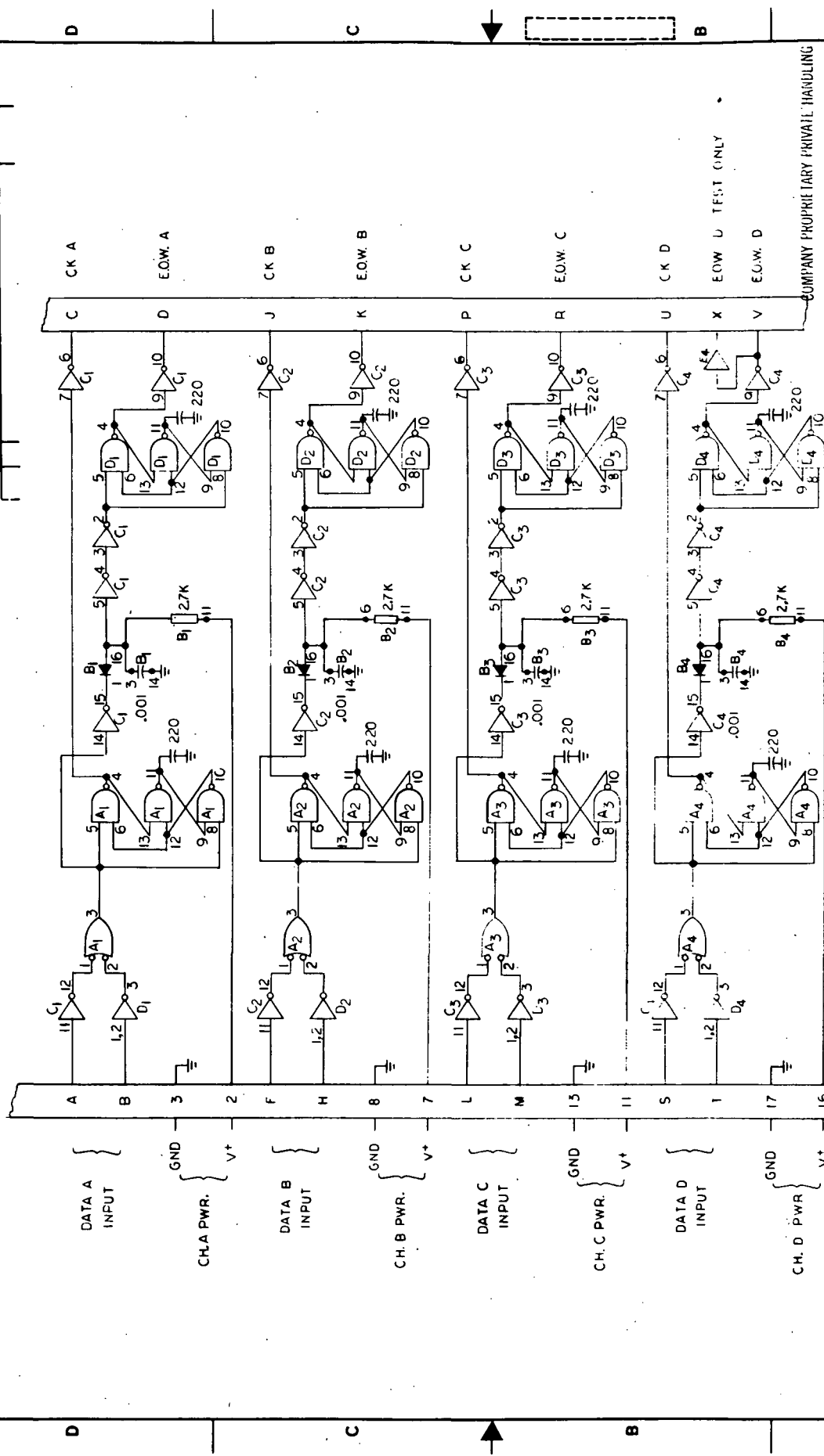
3.2.4 Supervisory Serial and Storage Registers/Modules 6 & 7; Locations C1, C2, D1, D2, C7, C8, C11 and C12

The purpose of these modules is to collect the zero's information and the one's information that is serially processed out of the serial voters and load it into parallel storage registers. The outputs of the storage registers, then, is parallel connected into decoders to decipher the particular commands.

The schematic of Figure 39 shows this hardware which is a straight forward design. "Zero's" and "one's" are processed through as "highs" for their existence and "lows" for their non-existence in the assigned register modules.

REVISIONS	DATE	APPROVED
1		
2		
3		
4		

ZONE	DESCRIPTION
1	
2	
3	
4	



ELECTRONIC COMMUNICATIONS, INC.
ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF RCA

ALL BREADBOARD-ALT. "B"

BOARD NO. 19

SYSTEM CLOCK GENERATOR

SIZE CODE IDENT NO. **C 00724**

SCALE **F/6 38**

SHEET

UNLESS OTHERWISE SPECIFIED

DIMENSIONS ARE IN INCHES

XX ± .02, XXX ± .010

TOLERANCES: ANGLES ±

WALL DIA. TOLERANCES

UP .001, .001 DIA. ± .010

THRU .001, .001 DIA. ± .010

REMOVE BURRS

BREAK SHARP EDGES .015 ± .010

MATERIAL

APPROVED

MAINT ENGR

APPROVED

FINISH

USED ON

APPLICATION

NEXT ASSY

A1-A4 D1-4 - CD4011 - 8EA.

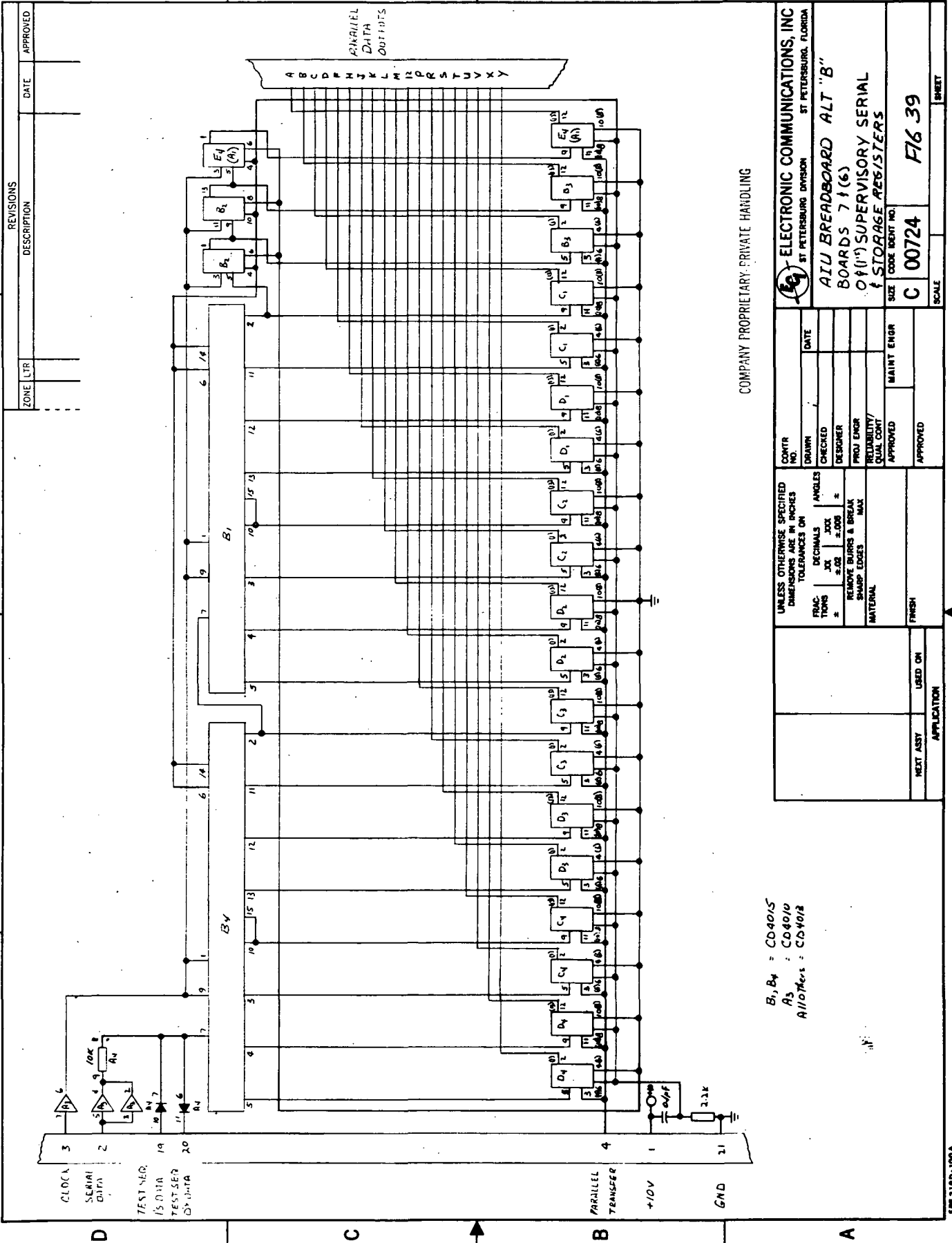
C1-4 - CD4009 - 4EA.

D1-4 - IN4148 - 4EA.

RES. - 27K 1/8W - 4EA.

CAP. - 220PF - 4EA.

E4 - CD4011 - 8EA.



COMPANY PROPRIETARY-PRIVATE HANDLING

B₁, B₄ = CD4015
 A₃ = CD4010
 All others = CD4012

ELECTRONIC COMMUNICATIONS, INC. ST PETERSBURG DIVISION ST PETERSBURG, FLORIDA	
ALU BREADBOARD ALT "B" BOARDS 71(6) OF (11) SUPERVISORY SERIAL & STORAGE REGISTERS	
SIZE CODE BENT NO. C 00724	SCALE F16 39
CONTR NO. DRAWN CHECKED DESIGNED PROJ ENGR RELIABILITY/QUAL CONT APPROVED	DATE DATE DATE DATE DATE DATE
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ON FRACTIONS DECIMALS ANGLES .XX .XXX .XXX .XX ±.02 .XXX ±.000 REMOVE BURRS & BREAK SHARP EDGES MAX MATERIAL	
NEXT ASST USED ON APPLICATION	FINISH APPROVED MAINT ENGR

3.2.5 Supervisory Parity Checker/Module No. 2; Locations C3, D3, C9, and C13

Figure 40 is a schematic of two parity trees each capable of verifying the parity of either the one's or zero's input depending on which is connected. The tree is oriented to test for odd parity and it is a straight forward design constructed of CD4030 exclusive-Or gates. (CMOS parity generators/checkers are not available at this writing).

3.2.6 Alternate "B" Supervisory Decoder/Module 12; Locations C4, D4, C10, D14

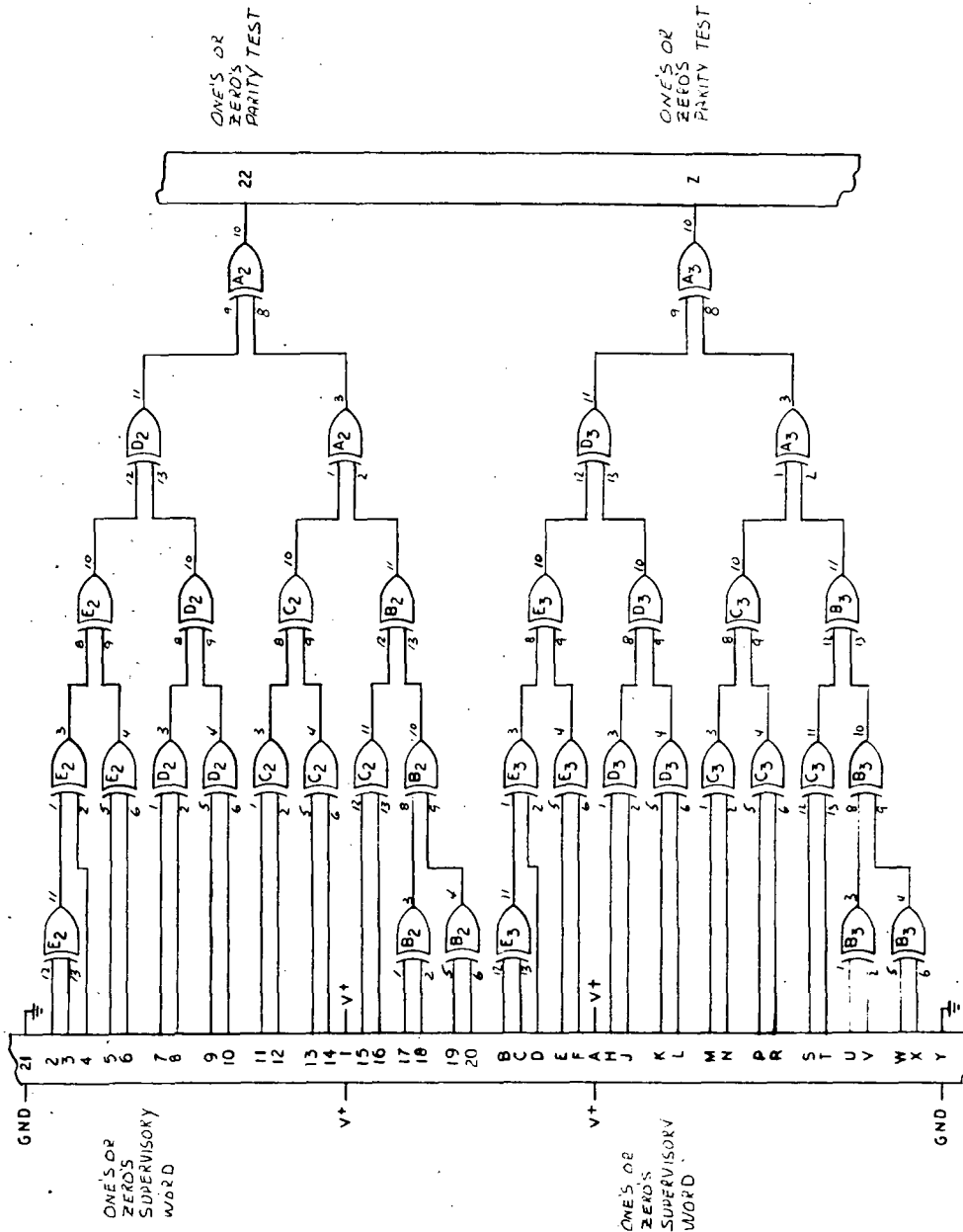
Each module has a "one's" and "zero's" decoder included as illustrated in Figure 41. Four BCD-to-Decimal decoders (CD4028) are employed with summing Nand gates as shown. The decoded zero's and one's are fed from here to the majority-voter summers which ultimately output the encoded command.

3.2.7 Single-Bit Majority Voter Summer/Module No. 3; Locations D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, A11, A12, A13, A14, B11, B12, B13 & B14

The logic in Figure 42 details the hardware required to vote and sum the decoded commands from the ones and zeros decoders. Two levels of majority voters are employed each voting a unique combination of two-out-of-three sets of inputs and then of the voted results. The output summers are stacked 8 wide such that the resultant output command can survive a minimum of two hard failures of the circuits feeding it. Two hard shorts to ground of two output gates, for instance, will tend to deteriorate the command level only 25% which is well within the switching parameters of the circuitry that the command feeds.

A majority voter summer is employed for each command. Although this circuitry appears highly complex, it is the easiest method of obtaining this level of failure immunity in an LSI configuration for which it was originally designed.

REVISIONS		DATE	APPROVED
ZONE	LTN	DESCRIPTION	
2			



COMPANY PROPRIETARY-PRIVATE HANDLING

ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA • A SUBSIDIARY OF RCA		CD 4030-10 EA 1e Siden Section	
BOARD NO. 2 SUPERVISORY PARITY CHECKER		SIZE CODE IDENT NO C 00724	
SCALE F/16 40		SHEET	
APPROVED MAINT ENGR		APPROVED	
NEXT ASSY USED ON		FINISH	
APPLICATION		APPLICATION	

REVISIONS

ZONE/LTR	DESCRIPTION	DATE	APPROVED

ALU BREADBOARD - ALT. "B"

BOARD # 12

1'S & 0'S DECODER

ELECTRONIC COMMUNICATIONS, INC.

ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF MCR

FIG 41

1'S DECODED COMMANDS

24 PIN IC

22k

+10

5-10GND

CD4028

CD4011

0'S DECODED COMMANDS

24 PIN IC

22k

+10

5-10GND

CD4028

CD4011

1'S REGISTER

24 PIN IC

22k

+10

5-10GND

CD4028

CD4011

0'S REGISTER

24 PIN IC

22k

+10

5-10GND

CD4028

CD4011

UNLESS OTHERWISE SPECIFIED

DIMENSIONS ARE IN INCHES

TOLERANCES: .XX ± .07, .XXX ± .010

ANGLES: 2

HOLE DIA TOLERANCES

1/16" ± .005

1/8" ± .005

1/4" ± .005

3/8" ± .005

1/2" ± .005

3/4" ± .005

1" ± .005

1 1/4" ± .005

1 1/2" ± .005

2" ± .005

2 1/2" ± .005

3" ± .005

3 1/2" ± .005

4" ± .005

4 1/2" ± .005

5" ± .005

5 1/2" ± .005

6" ± .005

6 1/2" ± .005

7" ± .005

7 1/2" ± .005

8" ± .005

8 1/2" ± .005

9" ± .005

9 1/2" ± .005

10" ± .005

10 1/2" ± .005

11" ± .005

11 1/2" ± .005

12" ± .005

12 1/2" ± .005

13" ± .005

13 1/2" ± .005

14" ± .005

14 1/2" ± .005

15" ± .005

15 1/2" ± .005

16" ± .005

16 1/2" ± .005

17" ± .005

17 1/2" ± .005

18" ± .005

18 1/2" ± .005

19" ± .005

19 1/2" ± .005

20" ± .005

20 1/2" ± .005

21" ± .005

21 1/2" ± .005

22" ± .005

22 1/2" ± .005

23" ± .005

23 1/2" ± .005

24" ± .005

24 1/2" ± .005

25" ± .005

25 1/2" ± .005

26" ± .005

26 1/2" ± .005

27" ± .005

27 1/2" ± .005

28" ± .005

28 1/2" ± .005

29" ± .005

29 1/2" ± .005

30" ± .005

30 1/2" ± .005

31" ± .005

31 1/2" ± .005

32" ± .005

32 1/2" ± .005

33" ± .005

33 1/2" ± .005

34" ± .005

34 1/2" ± .005

35" ± .005

35 1/2" ± .005

36" ± .005

36 1/2" ± .005

37" ± .005

37 1/2" ± .005

38" ± .005

38 1/2" ± .005

39" ± .005

39 1/2" ± .005

40" ± .005

40 1/2" ± .005

41" ± .005

41 1/2" ± .005

42" ± .005

42 1/2" ± .005

43" ± .005

43 1/2" ± .005

44" ± .005

44 1/2" ± .005

45" ± .005

45 1/2" ± .005

46" ± .005

46 1/2" ± .005

47" ± .005

47 1/2" ± .005

48" ± .005

48 1/2" ± .005

49" ± .005

49 1/2" ± .005

50" ± .005

50 1/2" ± .005

51" ± .005

51 1/2" ± .005

52" ± .005

52 1/2" ± .005

53" ± .005

53 1/2" ± .005

54" ± .005

54 1/2" ± .005

55" ± .005

55 1/2" ± .005

56" ± .005

56 1/2" ± .005

57" ± .005

57 1/2" ± .005

58" ± .005

58 1/2" ± .005

59" ± .005

59 1/2" ± .005

60" ± .005

60 1/2" ± .005

61" ± .005

61 1/2" ± .005

62" ± .005

62 1/2" ± .005

63" ± .005

63 1/2" ± .005

64" ± .005

64 1/2" ± .005

65" ± .005

65 1/2" ± .005

66" ± .005

66 1/2" ± .005

67" ± .005

67 1/2" ± .005

68" ± .005

68 1/2" ± .005

69" ± .005

69 1/2" ± .005

70" ± .005

70 1/2" ± .005

71" ± .005

71 1/2" ± .005

72" ± .005

72 1/2" ± .005

73" ± .005

73 1/2" ± .005

74" ± .005

74 1/2" ± .005

75" ± .005

75 1/2" ± .005

76" ± .005

76 1/2" ± .005

77" ± .005

77 1/2" ± .005

78" ± .005

78 1/2" ± .005

79" ± .005

79 1/2" ± .005

80" ± .005

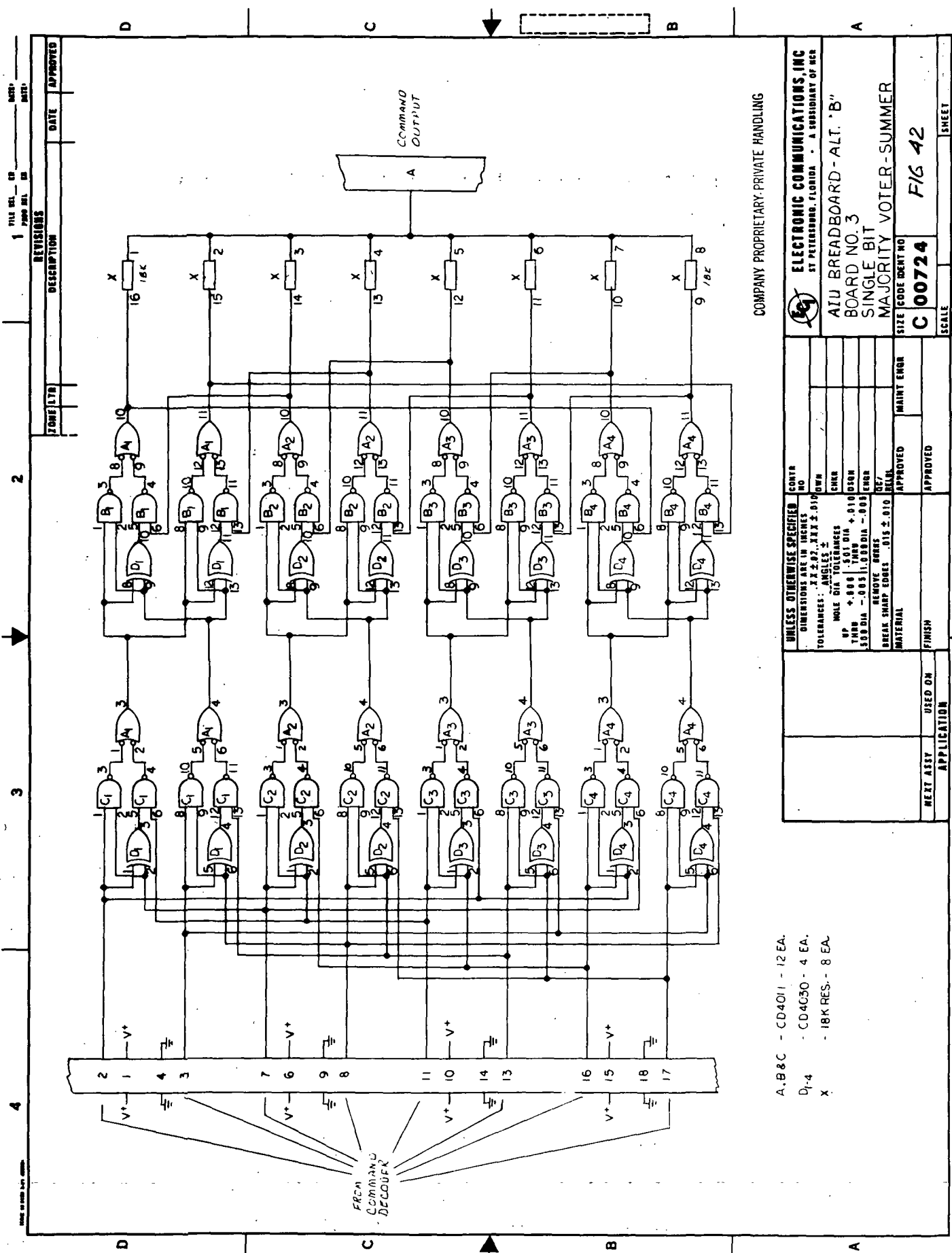
80 1/2" ± .005

81" ± .005

81 1/2" ± .005

82" ± .005

82 1/2" ± .005



A, B & C - CD4011 - 12 EA.
 D1-4 - CD4030 - 4 EA.
 X - 18K RES. - 8 EA.

COMPANY PROPRIETARY-PRIVATE HANDLING

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES: XX ± .01, XXX ± .010 ANGLES ± HOLE DIA TOLERANCES .015 ± .001 .010 DIA ± .010 .010 DIA ± .001 .010 DIA ± .001 BREAK SHARP EDGES .015 ± .010 MATERIAL FINISH		COSTS NO OWN CHN CSH ENR DE/
NEXT ASSY USED ON APPLICATION		APPROVED MAINT ENGR APPROVED
ELECTRONIC COMMUNICATIONS, INC. ST PETERSBURG, FLORIDA - A SUBSIDIARY OF RCA		SIZE CODE IDENT NO C 00724
ATU BREADBOARD - ALT. "B" BOARD NO. 3 SINGLE BIT MAJORITY VOTER-SUMMER		SCALE FIG 42

3.2.8 Data Word Serial & Storage Registers/Module No. 5; Locations A9, A10, B9 & B10

Two serial registers and two parallel storage registers comprise this logic in Figure 43 to complete this module. Each register processes a one's or a zero's serial stream of 8 bits of data word. Voting is done on each output bit to validate the data prior to entry to the rest of the system. The design is similar to the ISVC serial and storage registers in that they are clocked and loaded at the same time.

The data entry pins, pins 4 and 8, are buffered prior to feeding a diode/resistor network and the shift registers. This network permits the forced entry of simulation words by the test sequencer which overrides any input data that may be on line at the same time, Pin 11 carries the system clock and Pin 15 is the load pulse.

3.2.9 Data Bit Counters/Module No. 44; Location A7

The purpose of this logic is to verify that the entering data is of the proper bit count and that it has proper parity. If not, it will not be loaded into the storage registers. See Figure 44.

The EOW pulse resets the flip-flops C1, C2, C3, and C4 and the counters B1, B2, B3, and B4 prior to examining an incoming word. The flip-flop toggles with the occurrence of each "one" and has to end in the odd state. The counter counts the data stream and it has to count 8 data bits plus one parity bit. If these conditions are true, gates A1, A2, A3, and A4 permit the transmittal of the register load pulses. A logic differentiator resets the flip-flops and the



counter simultaneously.

3.2.10 Servo Amplifier & A/D Converter/Module No. 29; Locations E2, E4 & E6

(Reference paragraph 3.1.5; this module identical for both Alternate "A" and Alternate "B")

3.2.11 Eight-Bit Comparator ± 2 Bits/Module No. 58; Locations #1, E3 and E5

This module is identical to the one described in Para 3.1.6

3.2.12 Power Interrupt/Module No. 46; Locations H2, H3 and H4

(This logic is identical to that described in Para 3.1.7) See Figure 45.

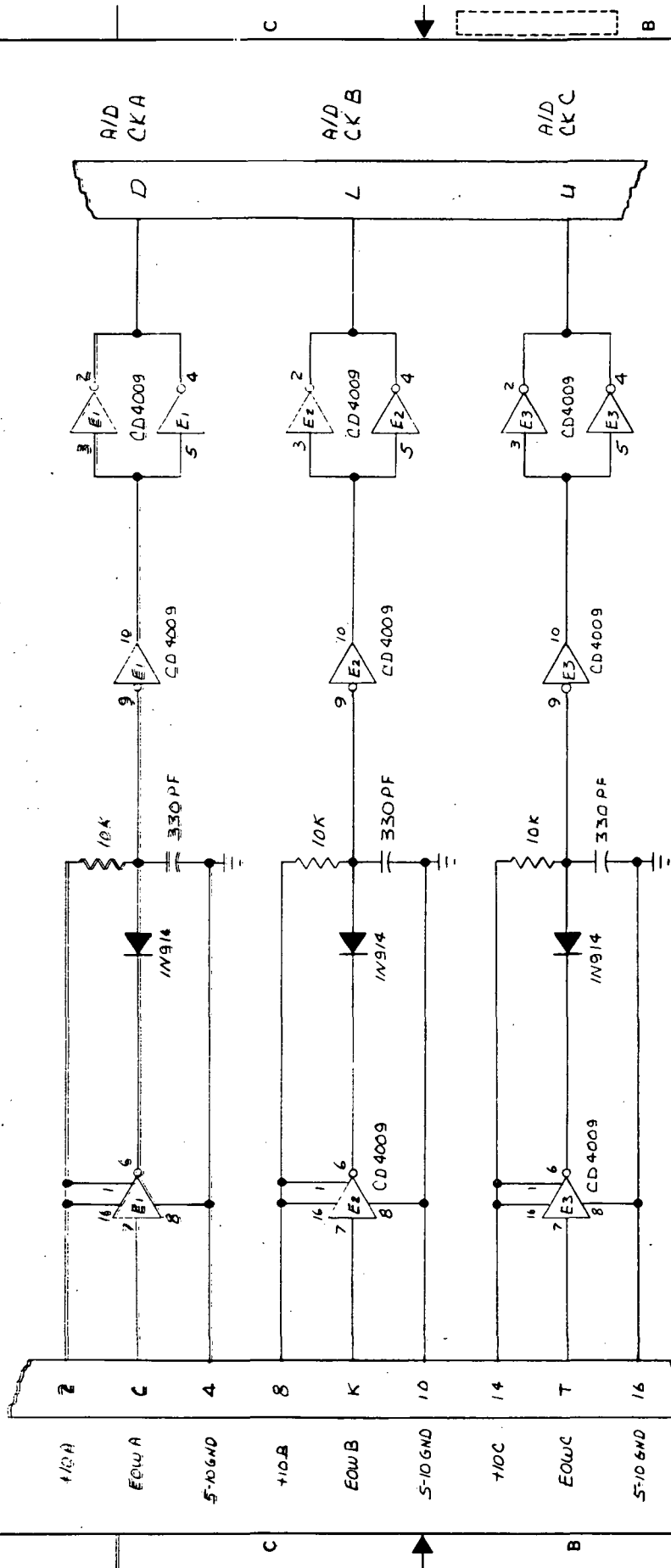
3.2.13 A/D Clocks/Module No. 31A; Location F8

The function of this module is to the END-OF-WORD pulse that occurs every 20 useconds and convert it to a clock that is usable by the A/D Converters. The end of word pulse is a narrow pulse used to load the incoming information into storage once a complete word has been received. This narrow pulse is stretched to approximately 2 usec in this logic before being sent on to the A/D converters. So the resultant clock is 50 KHZ with a pulse width of 2 usec. See the logic in Figure 46.

3.2.14 Test Sequencer System

The function of the test sequencer logic set is to respond to the supervisory command that dictates the unit to "self-test". The test sequencer then iterates through a

DATE	19	DATE	19
TIME	11:11	TIME	11:11
DESIGNATION	REVISION		
APPROVED	APPROVED		



ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF MCR	
AIU BREADBOARD-ALT. B" BOARD NO 31A A/D CLOCKS (50 KHZ)	
SIZE CODE IDENT NO C 00724	FIG 46
SCALE	SHEET
CONTR NO	CONTR NO
DIMENSIONS ARE IN INCHES TOLERANCES: .XX ± .02, .XXX ± .010	UNLESS OTHERWISE SPECIFIED
ANGLES ±	ANGLES ±
HOLE DIA TOLERANCES	HOLE DIA TOLERANCES
WIP + .005	WIP + .005
THRU + .005	THRU + .005
350 DIA - .005	350 DIA - .005
REMOVE POINTS	REMOVE POINTS
BREAK SHARP EDGES .015 ± .010	BREAK SHARP EDGES .015 ± .010
MATERIAL	MATERIAL
APPROVED	APPROVED
MAINT ENGR	MAINT ENGR
MEET ASSY	MEET ASSY
USED ON	USED ON
APPLICATION	APPLICATION

series of tests to verify the performance of redundant hardware and to verify the operation and linearity of the servo amplifiers.

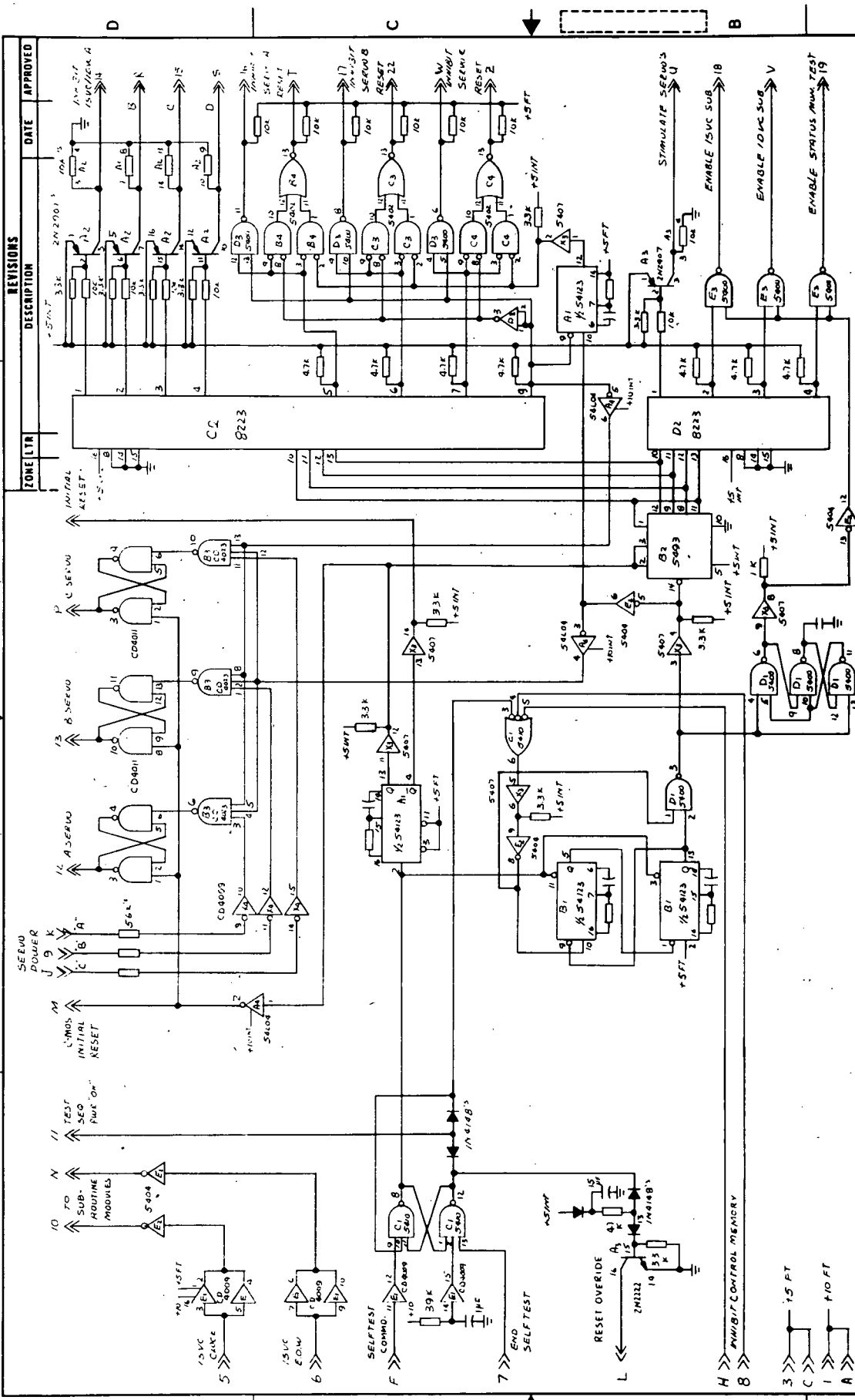
At the start of the test, the servo amplifiers are locked off and the ISVC and IDVC sections are inhibited in pairs while forced commands and forced data are inserted and compared to verify the operational performance of the circuitry to still provide legitimate information. Next, a "linearity" test is performed on each servo amplifier individually to verify that the servo amplifiers are performing within their 2-bit boundaries. Each amplifier is tested separately so that the hydraulic system is not effected since it will vote the response of two-out-of-three.

Following this test, the amplifiers are stimulated without entering a digital word into the data input and their automatic shutdown is recorded. Last, the status monitor sections are tested and then the encoded results of the complete test is loaded into the status monitor and reported to the test set or data terminal.

Table 3-2 is a truth table describing these events and it is also the truth table listing the controls stored in the control memory devices C2 and D2 of module no. 55.

3.2.14.1 Test Sequencer Control Memory/Module No. 55; Location F1

The logic of Figure 47 describes this central controller section of the test sequencer function. The information listed in Table 3-2 is stored and iterated from the control memory devices of this board.



REVISIONS		DATE	APPROVED
1	INITIAL	2/20/01	
2	RESET		
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			

ELECTRONIC COMMUNICATIONS, INC.
ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF RCA

AIU BREADBOARD - ALT "B"
BOARD NO. 55
TEST SEQ. CONTROL MEMORY

SIZE CODE IDENT NO
C 00724

SCALE
F16 47

SHEET

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES: ANGLES ± 0.1°
HOLE DIA TOLERANCES
UP +0.005, 0.01 DIA +0.010
DOWN -0.005, 0.01 DIA -0.005
REMOVE BURRS
BREAK SHARP EDGES 0.015 ± 0.010
MATERIAL
APPROVED
MAINT ENGR
FINISH

TEST ASST
USED ON
APPLICATION

3 >> 1.5 FT
C >> 1.0 FT
I >> 1.0 FT
4 >> 1.0 FT
21 >> 1.0 FT
Y >> 1.0 FT
D >> 1.0 FT
2 >> 1.0 FT
B >> 1.0 FT
20 >> 1.0 FT
X >> 1.0 FT

CONTROL MEMORY

MEMORY PINS	ADDRESS					Device C2							Device D2					
	14	13	12	11	10	B0	B1	B2	B3	B4	B5	B6	B7	b0	b1	b2		b3
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	Enable Stat Mon Subrtne.
0	0	0	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	Enable IDVC Subroutine
																		Enable ISVC Subroutine
																		Stimulate Servo Ampls.
																		Hard/Soft Servo Control
																		Inhibit Servo Ampl "C"
																		Inhibit Servo Ampl "B"
																		Inhibit Servo Ampl "A"
																		Inhibit ISVC/IDVC "D"
																		Inhibit ISVC/IDVC "C"
																		Inhibit ISVC/IDVC "B"
																		Inhibit ISVC/IDVC "A"

TABLE 3-2

ALTERNATE "B" SELF TEST CONTROL MEMORY PROGRAM

Enable Stat Mon Subrtne.
 Enable IDVC Subroutine
 Enable ISVC Subroutine
 Stimulate Servo Ampls.
 Hard/Soft Servo Control
 Inhibit Servo Ampl "C"
 Inhibit Servo Ampl "B"
 Inhibit Servo Ampl "A"
 Inhibit ISVC/IDVC "D"
 Inhibit ISVC/IDVC "C"
 Inhibit ISVC/IDVC "B"
 Inhibit ISVC/IDVC "A"

TABLE 3-2

ALTERNATE "B" SELF TEST CONTROL MEMORY PROGRAM

Latch C1 senses and stores the "self-test" command and when current is sunked into C1-12, the power supply for the test sequencer is turned "on". Normally, it is off to conserve power.

The transition of this latch causes the A1 one-shot to issue a reset pulse to normalize the conditions of the test sequencer during the "power-on" time. Also, the reset lines of the astable clock B1 are released and the astable is permitted to oscillate at 500 Hz. B2 is the address counter that increments the steps of the control memory. The memory outputs control the remaining portions of the sequencer to complete the tests. Two lines 8 and 9 carry inhibit controls to stop the incrementing clock during the ISVC and IDVC subroutines. Here, the tests cannot be completed within one clock period so the subroutine exercises its test to completion before returning control to the control memory logic.

The CMOS latches that are set by gates B3 memorize the automatic shutdown of the servo amplifier power interrupts.

Transistor A3 is used as a switch to stimulate the servo's with an analog voltage when the data inputs are held at zero.

3.2.14.2 ISVC Test Subroutine Logic/Module No. 51; Location F2

This logic iterates all supervisory commands through the ISVC section while permutating 31 encoder-inhibit combinations to verify its redundancy as well. See Figure 48. Latch C2/C3 holds the subroutine on until it is reset by the word generator/counter A2. A shift register B1 is used to generate the timing and

control for generating the clock, load and error-read pulses required for this test. Shift register A3/X3 is loaded with each supervisory command and shifts the words out to the forcing drivers located at E2. Device X1 provides parity for each word. Devices X4 and D4 with the peripheral gating logic shown with them, provide the inhibit patterns for the ISVC decoder sections which are iterated for each test word sent out. Each majority-voter-summer is interrogated for each word and for each test of that word by the read-pulse that outputs on pin 11.

Device E1 provides a re-triggerable state to inhibit the control memory while these tests or the IDVC tests are being conducted. The timing and control for these tests are derived from the systems clock and end-of-word pulses. The clock derivation sections of ISVC are permitted to continually operate even though the word information is forced to be what the test routine makes it. In this way, proper system timing is maintained.

3.2.14.3 ISVC Subroutine Test Encoder/Module No. 52; Location F6

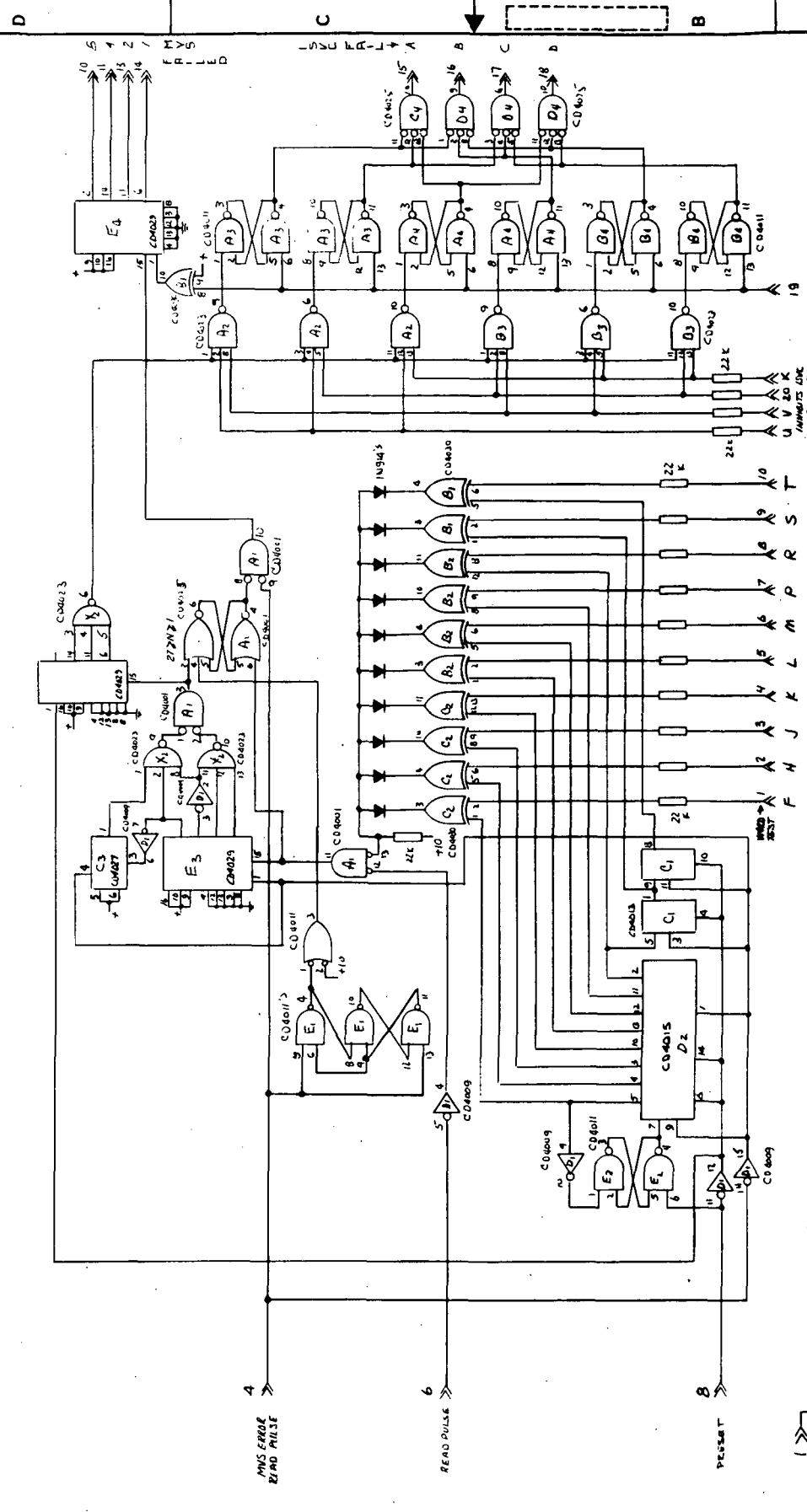
This logic, shown in Figure 49, accounts for the ISVC failures. With each test combination issued by the subroutine logic a read pulse permits interrogation of the results of that test.

Shift register D2 and C1 establish which word is under test and that word is compared with the state of the corresponding majority-voter-summer. A set of exclusive Or's are Ored together by a diode set to insure that the word under test did get issued by the majority-voter-summer.

1 FILE REL. IN
2 PART REL. IN
3 DATE:
4 DATE:

REVISIONS

ZONE LTR	DESCRIPTION	DATE	APPROVED
----------	-------------	------	----------



- 7400 - 5
- 7401 - 1
- 7402 - 1
- 7403 - 1
- 7404 - 1
- 7405 - 1
- 7406 - 1
- 7407 - 1
- 7408 - 1
- 7409 - 1
- 7410 - 1
- 7411 - 1
- 7412 - 1
- 7413 - 1
- 7414 - 1
- 7415 - 1
- 7416 - 1
- 7417 - 1
- 7418 - 1
- 7419 - 1
- 7420 - 1
- 7421 - 1
- 7422 - 1
- 7423 - 1
- 7424 - 1
- 7425 - 1
- 7426 - 1
- 7427 - 1
- 7428 - 1
- 7429 - 1
- 7430 - 1
- 7431 - 1
- 7432 - 1
- 7433 - 1
- 7434 - 1
- 7435 - 1
- 7436 - 1
- 7437 - 1
- 7438 - 1
- 7439 - 1
- 7440 - 1
- 7441 - 1
- 7442 - 1
- 7443 - 1
- 7444 - 1
- 7445 - 1
- 7446 - 1
- 7447 - 1
- 7448 - 1
- 7449 - 1
- 7450 - 1
- 7451 - 1
- 7452 - 1
- 7453 - 1
- 7454 - 1
- 7455 - 1
- 7456 - 1
- 7457 - 1
- 7458 - 1
- 7459 - 1
- 7460 - 1
- 7461 - 1
- 7462 - 1
- 7463 - 1
- 7464 - 1
- 7465 - 1
- 7466 - 1
- 7467 - 1
- 7468 - 1
- 7469 - 1
- 7470 - 1
- 7471 - 1
- 7472 - 1
- 7473 - 1
- 7474 - 1
- 7475 - 1
- 7476 - 1
- 7477 - 1
- 7478 - 1
- 7479 - 1
- 7480 - 1
- 7481 - 1
- 7482 - 1
- 7483 - 1
- 7484 - 1
- 7485 - 1
- 7486 - 1
- 7487 - 1
- 7488 - 1
- 7489 - 1
- 7490 - 1
- 7491 - 1
- 7492 - 1
- 7493 - 1
- 7494 - 1
- 7495 - 1
- 7496 - 1
- 7497 - 1
- 7498 - 1
- 7499 - 1
- 7500 - 1

ELECTRONIC COMMUNICATIONS, INC.
ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF RCA

ACTUATOR INTERFACE BOARD
AUTOMATE "B" - BD # 52
ASIC TEST ENCODER

SIZE CODE ORT NO
C 00724

FIG 49

SCALE

SHEET

UNLESS OTHERWISE SPECIFIED	CONT.
DIMENSIONS ARE IN INCHES	NO
TOLERANCES: .XX ± .01, .XX ± .01	MIN
ANGLES: ± .01	MIN
HOLE DIA TOLERANCES	MIN
PP .006 ± .010	MIN
TURN .006 ± .010	MIN
500 DIA .005 ± .010	MIN
BREAK SHARP CORNERS	MIN
MATERIAL	MIN
FINISH	MIN
MAINT ENGR	MIN
APPROVED	MIN
USED ON	MIN
APPLICATION	MIN
WERT ASSY	MIN

Two types of errors are recorded; those related to a failed ISVC channel and those related to a failed decoder or majority-voter-summer set.

Counter E3/C3 records the number of errors that occur during the iterations of decoder-inhibit tests. Latch A1 registers any number between one and twenty seven. A counter decoded by X2 registers a continuous failure of a word set including the encoder-inhibit operations. Counter E4 stores the number of decoder/MVS failures and reports them in terms of a binary number in increments of 1, 2, 4, and 8. A small number of failures could be considered transient failures caused by noise, etc. A large number of failures that are repeatable would be indicative of hardware problems requiring maintenance. It would be difficult to include sufficient diagnostic hardware to pin point the exact failed hardware as apposed to the present method of indicating the existence of a problem that is not normally seen because of redundancy compensation.

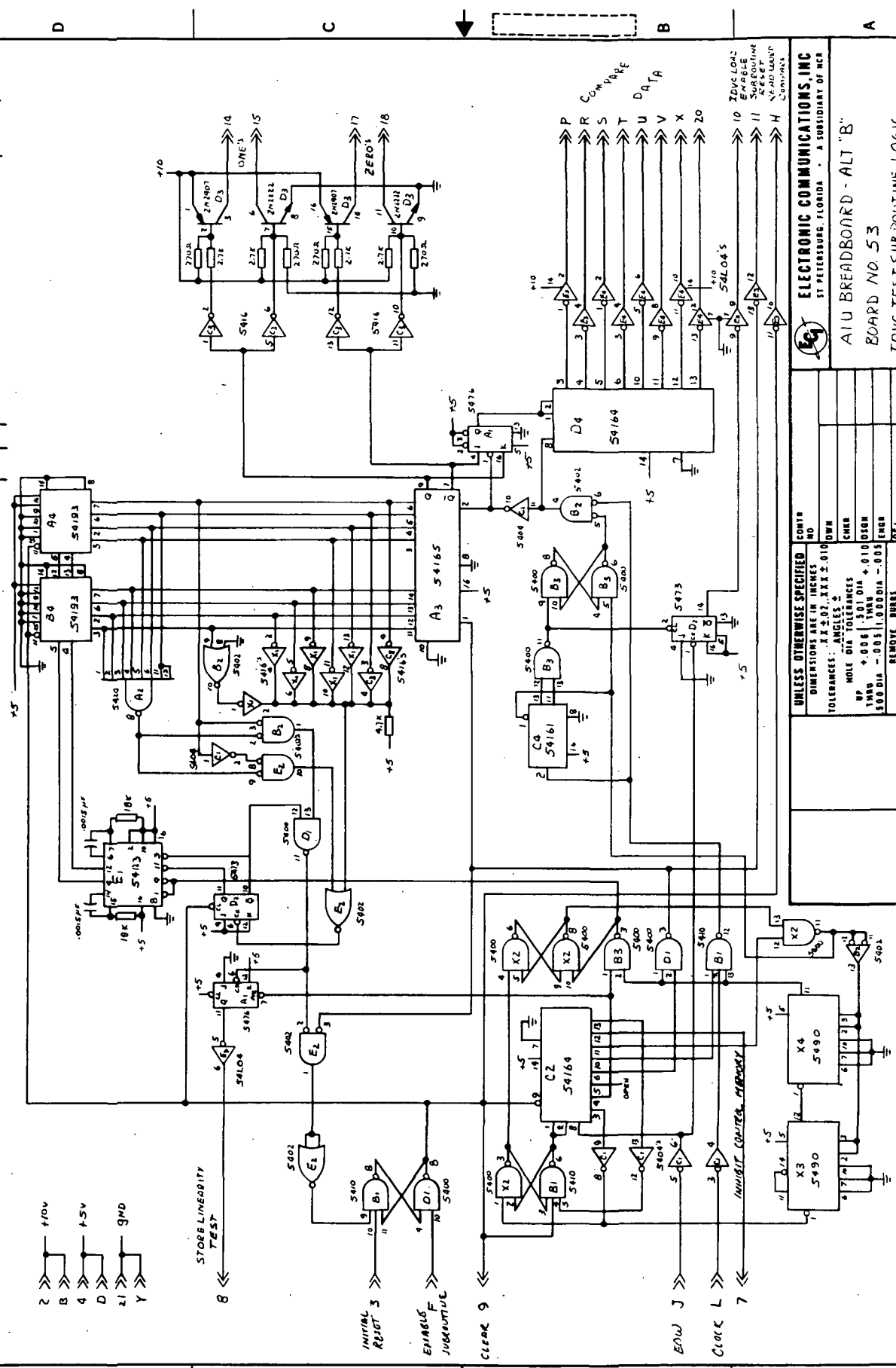
Where there is a specific channel failure in ISVC, the failed tests are recorded in a set of latches A3, A4 and B4, and then are decoded in gates C4 and D4 to indicate which channels have failed.

Both the MVS failures and the channel failures are reported as part of the status word.

3.2.14.4 IDVC Test Subroutine Logic/Module No.53; Location F3

Figure 50 shows the logic for the IDVC word test.

REVISIONS		DATE	APPROVED
1	FILE REL. 18		
2	PROD REL. 18		
DESCRIPTION		DATE	APPROVED
1	ZONE LTR		



ELECTRONIC COMMUNICATIONS, INC.		ST. PETERSBURG, FLORIDA - A SUBSIDIARY OF NCR	
AIU BREADBOARD - ALT "B"		BOARD NO. 53	
IDVC TEST SUBROUTINE LOGIC		SIZE CODE IDENT NO	
C 00724		FIG 50	
NEXT ASST		USED ON	
FINISH		APPROVED	
MATERIAL		APPROVED	
BREAK SHARP EDGES .015 ± .010		APPROVED	
REMOVE BURRS		APPROVED	
HOLE DIA TOLERANCES		APPROVED	
HOLE DIA TOLERANCES		APPROVED	
HOLE DIA TOLERANCES		MAINT ENGR	SHEET

This logic operates similarly to the ISVC system in that it continually generates simulated words and forces them into the front end of the IDVC section.

Latch B1/D1 is set when the subroutine is commanded to start. Shift register C2, counter X3/X4 and gates X2, B3, D1 and B1 develop the timing and control such that the logic will output a new word in every other 20usec time frame.

Counter B4 and A4 develop the sequential data words which are loaded into shift register A3 and shifted out through the forcing driver transistors. Each data iteration actually covers the data word spectrum twice. The words start at one-bit greater than zero or at the word (10000001). The counter counts up to all ones (11111111) and then reverses direction and counts down to all zeros (00000000) at which time it again reverses direction and counts up to one bit less than zero or the word (01111111). Here, it stops until the next test iteration.

During the IDVC channel tests, each word is strobed and compared with the actual word sent out. Shift register D4 stores the last word out and pin H carries the read pulse.

During the linearity test, the total number of accumulated errors are stored and the storage pulse for this test output on pin 8.

3.2.14.5 Test Sequencer 8 Bit Comparator/Module No. 56; Location F4

This module is used specifically for the servo linearity test. Each of A/D outputs of the servo amplifier monitors are fed to this module to a multiplexer input. See Figure 51. The amplifier under test determines which information will be multiplexed in. The comparator portion is virtually identical to the 8-bit ± 2 -bit comparators used in the servo amplifier verification system. The output of this comparator,

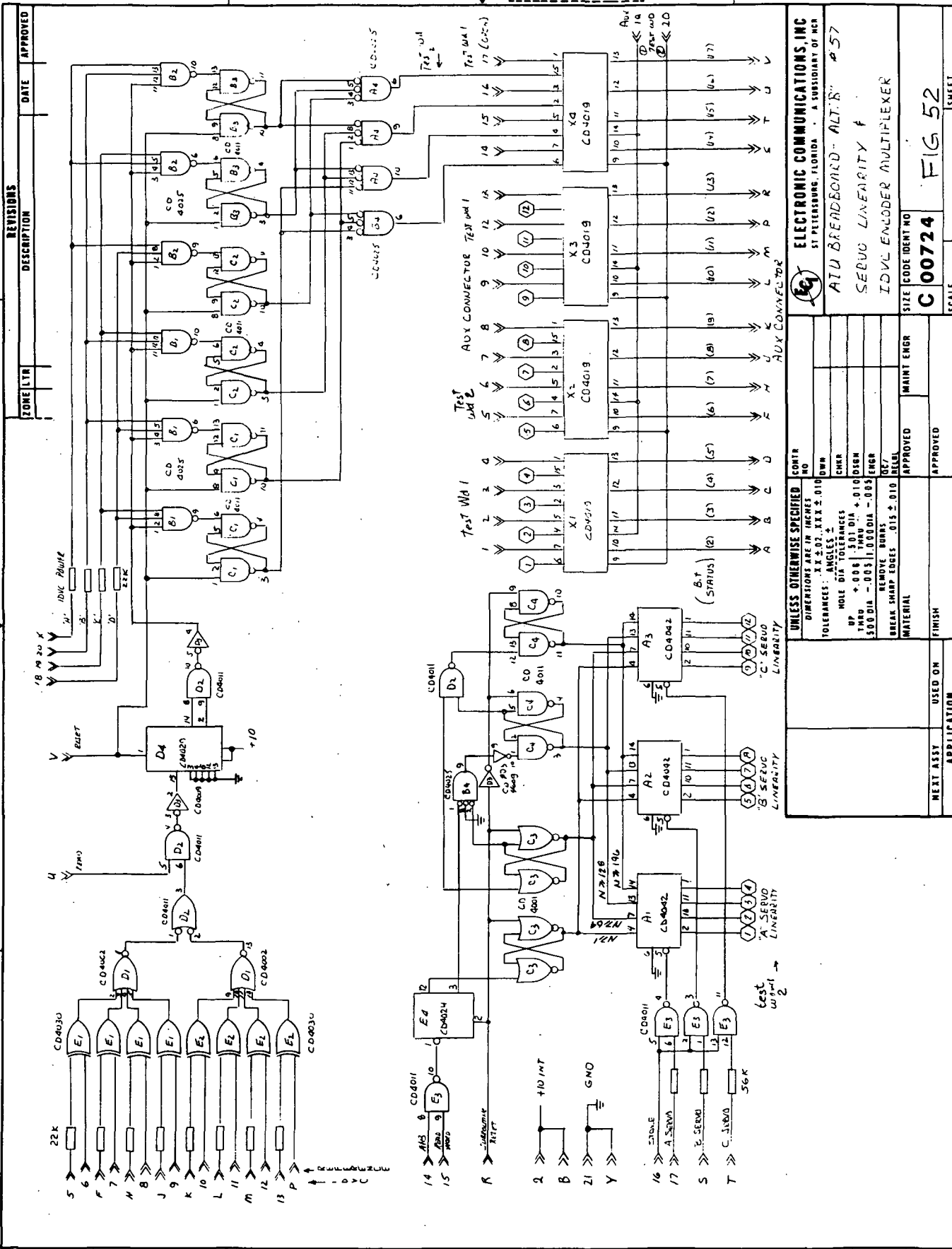
however, is simple $A=B$ or $A \neq B$ which outputs on pin T from an interface transistor that brings the logic level back to 10 volts.

3.2.14.6 Servo Linearity & IDVC Encoder Multiplexer

The logic in Figure 52 details this function. Devices E1 and E2 compare the IDVC information with the test word on each entry and the errors are counted by D4 and recorded by the appropriately enabled latches C1, C2 and B3. Gates A4 and B4 decode the failed channel and the multiplexers X1, X2, X3 and X4 select the word that the information will be reported on. The servo linearity test is recorded by reading the output of the 8-bit comparator and counting the errors in E4. Gates B4, D2 and latches C3 and C4 record the errors count in A1, A2 and A3. These are loaded in accordance with the servo under test and the load pulse which is selected by gates E3.

The outputs are weighted to account for error numbers of 1, 64, 128 and 196 respectively. They are weighted this way because of the 512 comparisons made during the linearity test. The error number is merely a gauge as to the amplifiers performance and is not intended as a criteria for rejection.

Both status words which are issued as a result of the test sequencer operation are multiplexed through the X devices. The inputs are labeled as to which information is concerned with which word. The auxillary connector pins A through V correspond to bits 2 through 17 of the status words as displayed on the test set.



				UNLESS OTHERWISE SPECIFIED		CONTR NO				ELECTRONIC COMMUNICATIONS, INC. ST PETERSBURG, FLORIDA - A SUBSIDIARY OF NCR	
				DIMENSIONS ARE IN INCHES						ATU BREADBOARD - ALT. 8 - #57	
				TOLERANCES: .XX ± .02, .XXX ± .010						SERVO LINEARITY #	
				ANGLES ±		DOWN				IDVL ENCODER MULTIPLEXER	
				HOLE DIA TOLERANCES		CHAM					
				UP							
				THRU							
				300 DIA							
				REMOVE BURNS		ENGR					
				BREAK SHARP EDGES .015 ± .010		OK?					
				MATERIAL		REEL					
						APPROVED		MAINT ENGR		SIZE (CODE IDENT NO)	
				FINISH		APPROVED				C 00724	
				NEXT ASSY						FIG 52	
				APPLICATION						SCALE	
										SHEET	

3.2.14.7 Status Monitor Subroutine Logic/Module No.54; Location F7

See Figure 53. This logic inhibits the Polar RZ Modulator so that no activity gets out to the bus during the status monitor hardware test. It initiates a "test word request" which is a preprogrammed test word of alternate ones and zeros. This request dictates that the status monitor issue the test word as if it were requested via the data bus or test set. At the same time that the status monitor processes the test word, the subroutine logic issues a similar word from shift register D3. The E4 exclusive Or's perform a direct bit-for-bit comparison on each of the words being issued from the four channels of the status monitor, with exception to the polar RZ modulations. If errors occur they are stored in latches B1, B2, B3 and B4 and at the end of the test, the information is shifted up to latches C4 and D4. The shift register C2, develops the timing for this scheme and it also develops the timing for the load/send commands to output the test sequencer results plus initiate the "end-of-test" signal shuts the test sequencer function off and powers it down. The one-shot A2 is triggered to provide approximately 20 seconds between issued test result words. This time is offered to allow the test set operator to be able to manually record the first word information before the second word erases it and is displayed. The timing can be altered for automatic operation where the two words are sent in successive 20usec time frames.

3.2.15 Status Monitor Timing and Control/Module No. 45; Locations F9, F10, F11 and F12 (See Figure 54)

(This logic is identical to that described in Para 3.1.19) Four modules are employed

to handle the timing and control of the four channels of the status monitor function.

3.2.16 Status Monitor Multiplexer/Module No. 15; Locations E9, E10, E11, E12, G9, G10, G11 and G12. (See Figure 55)

Each multiplexer module is capable of multiplexing 6 8-bit words to one 8-bit output. With the four modules, then, 6 16-bit words are selected and multiplexed into four identical status monitor channels.

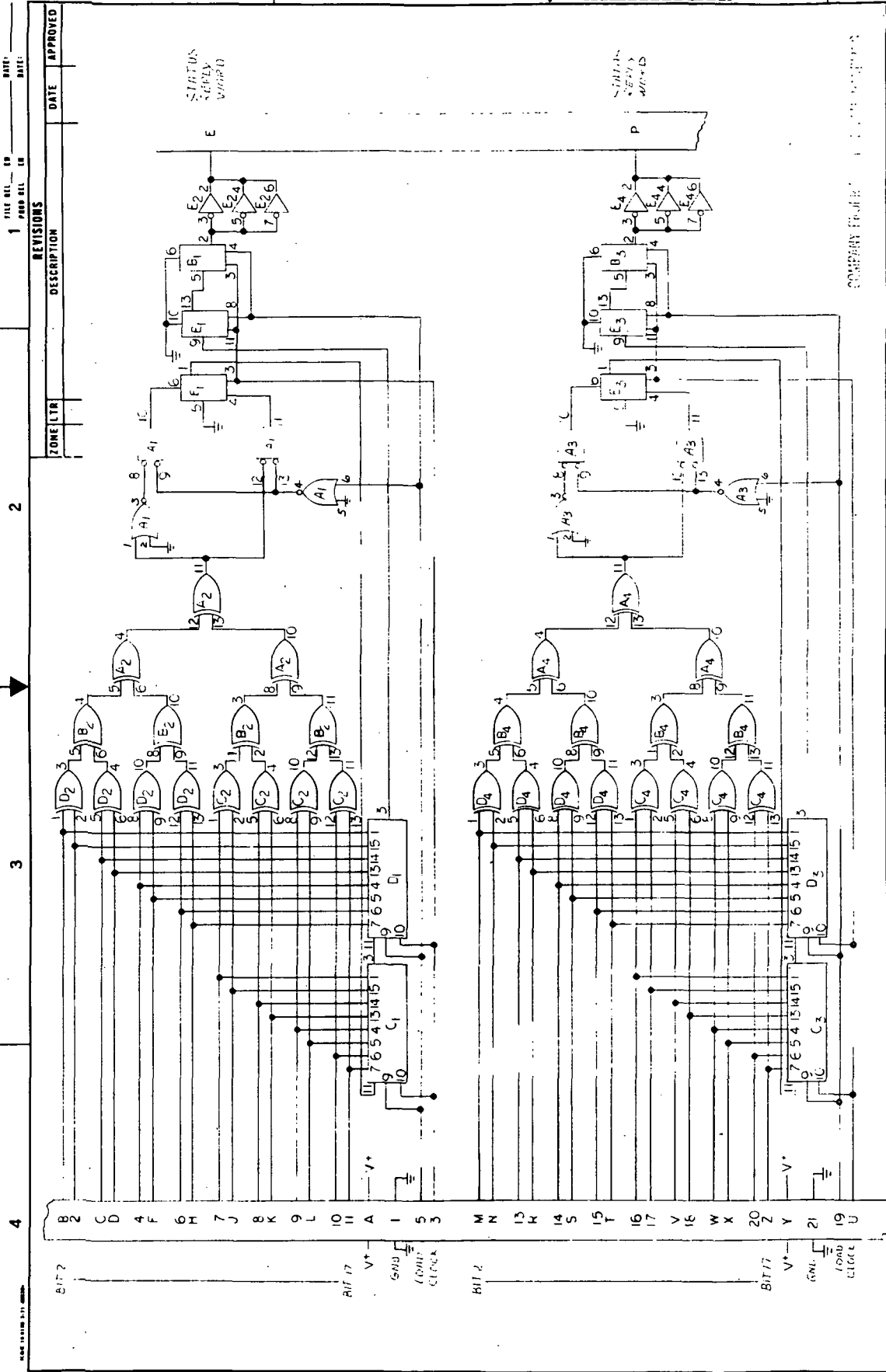
The design configuration is straight forward employing the CD4019 selector gates with CD4025's used to further reduce the selections to eight outputs total.

3.2.17 Status Monitor Parity Generator and Shift Register/Module No. 14; Locations E13 and G13

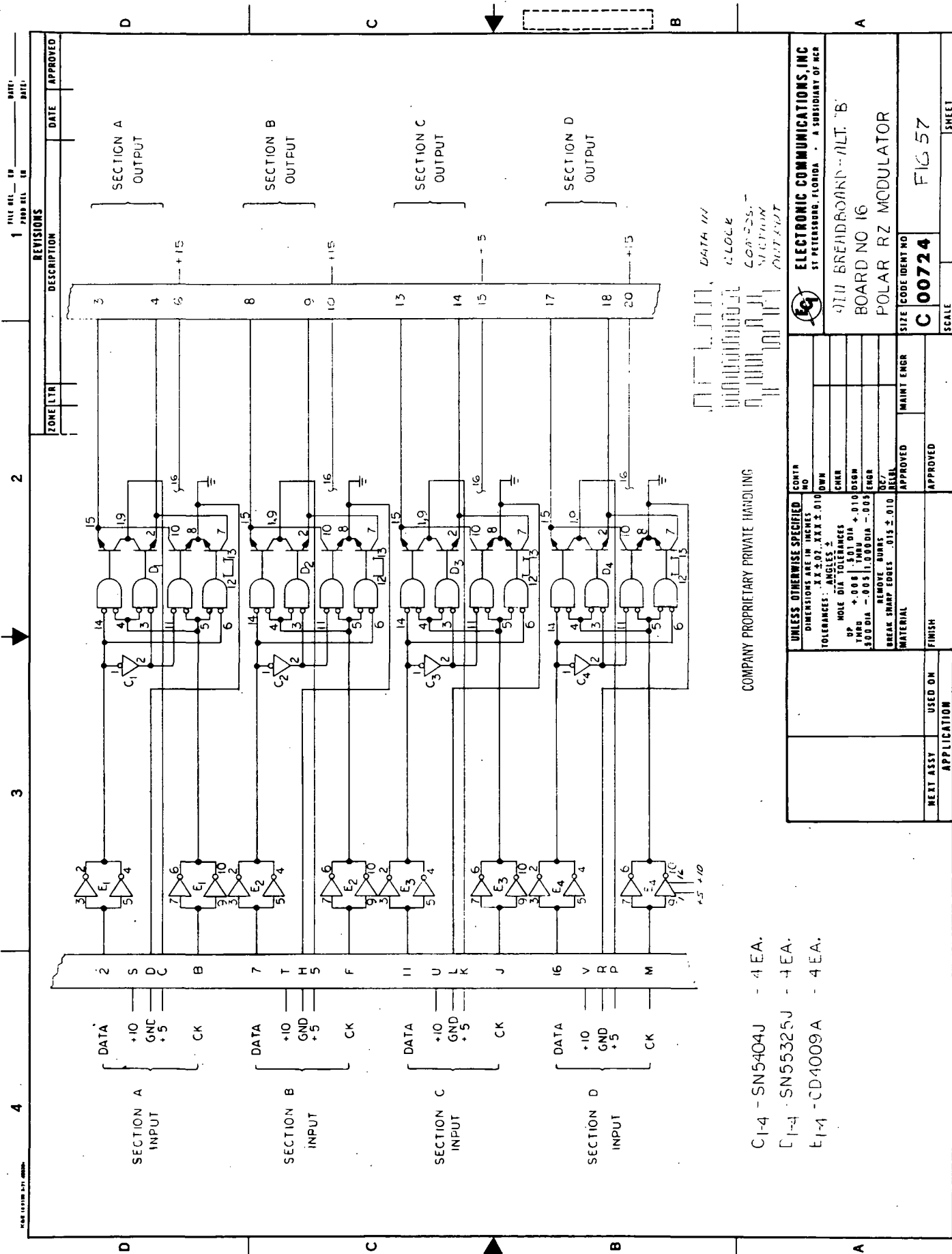
Figure 56 describes this logic which is comprised to two channels of parity test and shift register storage each. Devices C1 and D1 are loaded with the sixteen bit word and device E1 is loaded with the parity bit which is generated by the parity tree D2, C2, B2 and A2. The first three bits of each word are fixed in devices E1 and B1. Each status monitor word is loaded and shifted out to the polar RZ modulators identically in all four sections of the two modules.

3.2.18 Polar RZ Modulator/Module No. 16; Location F13

Figure 57 illustrates the four channel polar RZ modulators required for the status monitor system. Each modulator is configured using the SN55325 core memory driver device which is well suited to this modulator application. The schematic illustrates the data and clock patterns and the composite output that is put on the line.



REVISIONS 1 FILE REL. 10 2 1000 REL. 10 3 4		DATE APPROVED
DESCRIPTION ZONE LTR		DATE APPROVED
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES: .XX ± .02, .XXX ± .010 ANGLES: ± HOLE DIA. TOLERANCES DP +.008, -.001 DIA. ±.010 THRU +.008 THRU -.005 ENGR 500 DIA. -.005 THRU -.005 ENGR REMOVE BURRS BREAK SHARP EDGES .015 ± .010 REEL MATERIAL		CONTR NO DWN CKR DISM ENGR REL APPROVED APPROVED
ELECTRONIC COMMUNICATIONS, INC. ST. PETERSBURG, FLORIDA • A SUBSIDIARY OF NCR		SCALE SHEET
ALU BREADBOARD-ALT. 'B' BCARD NO. 14 STATUS MONITOR PARITY GEN. & SHIFT REGISTER		SIZE (CODE IDENT NO) C 00724
NEXT ASSY USED ON APPLICATION		FINISH APPROVED
A ₁ A ₂ A ₃ A ₄ B ₁ B ₂ B ₃ B ₄ C ₁ C ₂ C ₃ C ₄ D ₁ D ₂ D ₃ D ₄ E ₁ E ₂ E ₃ E ₄ F ₁ F ₂ F ₃ F ₄ G ₁ G ₂ G ₃ G ₄ H ₁ H ₂ H ₃ H ₄ I ₁ I ₂ I ₃ I ₄ J ₁ J ₂ J ₃ J ₄ K ₁ K ₂ K ₃ K ₄ L ₁ L ₂ L ₃ L ₄ M ₁ M ₂ M ₃ M ₄ N ₁ N ₂ N ₃ N ₄ O ₁ O ₂ O ₃ O ₄ P ₁ P ₂ P ₃ P ₄ Q ₁ Q ₂ Q ₃ Q ₄ R ₁ R ₂ R ₃ R ₄ S ₁ S ₂ S ₃ S ₄ T ₁ T ₂ T ₃ T ₄ U ₁ U ₂ U ₃ U ₄ V ₁ V ₂ V ₃ V ₄ W ₁ W ₂ W ₃ W ₄ X ₁ X ₂ X ₃ X ₄ Y ₁ Y ₂ Y ₃ Y ₄ Z ₁ Z ₂ Z ₃ Z ₄		FIG 56



3.2.19 Alternate "B" Power Supplies A, B, and C/Module No. 49; Locations H7 H9 and H11

These power supplies are straight forward switching inverter designs to convert the incoming 28VDC $\pm$ 4VDC to the several voltages required by the system. Both I.C. and transistor regulators are employed as shown in Figure 58.

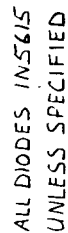
The supplied 10VDC and 5VDC, as shown, is configured to track so that the 10VDC supply is always higher than the 5 VDC even during the turn-on and turn-off cycles.

The 15 volt and 30 volt supplies are for the servo amplifiers and A/D converters in the system. A separate supply is provided for each of the system channels; 4 redundant channels in the input/output and three channels of servo. Power supplies A, B, and C serve three of the four I/O channels and the three servo channels.

3.2.20 Alternate "B" Power Supply D/Module No. 50; Locations H13

This power supply is similar in design approach as the A, B and C supplies except that it does not supply 30V and the 15 V required by the servos. Instead, it supplies the fourth channel of the I/O channels plus it provides a controllable supply for the test sequencer function. See Figure 59. A 10VDC @360ma and a 5VDC @2.8A are provided for the test sequencer and a shutdown control inputs on pin 5. These supplies are normally disabled until the test sequencer is commanded to operate.

It can be seen that the supplies do require rather high input and output power levels. This would not be the case in the final design configuration which would be implemented in CMOS/LSI throughout. Much of this breadboard configuration was an experiment in design configuration feasibility and, to expedite these design



10

configurations, TTL was employed since it had a more versatile family at the time. Consequently, the TTL hardware required more power, and of course, larger power supplies. The design, now, can immediately be converted to CMOS/LSI from the discrete design which would result in considerably less power required. The only exercise remaining would be to redesign the power supplies for smaller size but with the same controllability.

4.0 CONCLUSIONS

The Actuator Interface Unit(s) development program provided several conclusions that are worthy of consideration. Some of these conclusions are interrelated and some will deal with general practices relating these types of development efforts.

1. The design concept of Alternate "A" which was originated at Huntsville, Alabama as the original "look and switch" concept has been proven to be feasible and can be implemented. The breadboard implementation is impractical even if the discrete components could be densely packaged. However, the LSI implementation, which was the intended end item, is definitely practical both in terms of size and power consumption plus reliability.
2. Alternates "A" and "B" breadboards were intended to do two things; prove the design concept and yield practical circuit designs in CMOS. As stated, the program accomplished this intent. However, during the construction phases of the hardware, it became obvious that some of the circuit designs were impractical in the discrete format. Circuits such as the Majority Voter Summer which would be simple in the LSI arena became highly complicated in the discrete fashion - and expensive.

Here, gate hardware, cheap in LSI, was employed at the maximum to avoid the necessity of passive components which cannot be integrated within an LSI chip.

Other circuits employed techniques which are not exact schematic conversions to LSI because of the discrete hardware availability. A five-input Or function, for instance, had to be fabricated from many more gates in discrete format than would have been necessary in LSI. This problem is the opposite extreme of the one stated above which used gates in place of the apparent simplicity of passive devices.

Because of these kinds of factors, it is not practical to hardware breadboard an LSI electrical design. Not mentioned is the obvious propagation problems that arise in the large physical architectures of the breadboard but would not exist in the small LSI package. The problems are easily solvable but it occasionally required hardware that causes the schematic to not be directly convertible. The conclusion, then, is that there has to be a more practical method of proving out an electrical design intended for LSI production. And there is.

Since the A. I. breadboard programs were undertaken, several companies including ECI's parent company, NCR, have developed simulation programs which library catalog CMOS devices according to their logic function and their characteristics. Whole logic designs can be entered as if they were constructed with these devices and test programs written that simulate actual inputs and measure resultant outputs that do not

suffer the imposing characteristics of discrete architecture.

Further failure modes can be iterated to determine the realistic logic performance under these conditions. This technique would be ideal for verifying redundant configurations that would ultimately be fabricated in LSI form.

3. ECI undertook the breadboard construction with the intent to eliminate the many problems that occur using the very large logic cards and large amounts of back-plane wiring. This is a common fabrication technique of complex breadboards because they are easily altered and expanded as required. However, because of the wiring density that ultimately occurs over the back-plane(s), noise immunity and cross-coupling of signals often become serious enough that the boards have to be relayed out to make the system perform. The system ECI employed was to break the system down into modular designs using the same wire-wrap technique but each module would perform as a separate and often as a duplicate function. This is seen in the discussions of these various modules included in this report.

Basically, the idea was fruitful. Modular interchangeability became an excellent tool for troubleshooting and the systems electrical design was more orderly when thought of as a set of functional blocks. Further, the physical back-plane wiring was less complex than if the breadboard had been constructed on a single plane. The cards selected were easily adapted to alteration for both i. c. 's and discrete devices.

However, one single problem continually threaded itself throughout the fabrication function and that was edge connector reliability. Because the cards were extracted over and over, the plating on the connectors wore thin and would oxidize easily. This altered the contact resistance sufficiently that many TTL crossovers simply wouldn't work. The CMOS crossovers were virtually unaffected because these devices do not have current sinking/sourcing input criteria. If the crossover frequency was high, say, 1 MHz, then often the pulse characteristics would change because of the CMOS input capacity. However, even this kind of change did not always alter the system performance.

A cure for this connector problem was not found during the course of the program except that keeping the connectors clean was necessary. Often, this would become troublesome because too much time was spent on seeking the "latest fault point" that was interfering with the more subtle problems concerning actual systems performance.

One conclusion that was drawn was that the edge-connector board that was selected for the program was acceptable as a commercial quality device as long as it was not over used or handled excessively. Unfortunately, these cards had to be handled "excessively" since they were experimental designs. They are not, therefore, acceptable as breadboard devices.

Now, this is not a problem that would happen on the single-plane breadboard scheme. Here, virtually all interconnects are done right

to the wire-wrap sockets which house the i. c. devices. Connectors per se are at a minimum.

Although the solution to this problem is not immediately at hand to be applied to the A.I. breadboard systems, ECI is presently investigating the feasibility of applying certain lubricants to the edge connector such as silicon grease. Preliminary testing has revealed that the connector pressure is sufficient to "wipe" the contact point clean for good contact but, at the same time, the grease will migrate to protect the remaining portions of the contact from corrosion build-up. The predictability of the effects of the lubricant conductivity has not yet been determined and, of course, this would adversely effect the cross-contact of devices - especially CMOS since it has high impedance input characteristics.

A supplementary report will follow this final report when the solution is resolved such that the A.I. breadboard systems can be made more functionally reliable.

APPENDIX A

GENERAL DERIVATION OF SECOND ORDER RUNGE - KUTTA

APPENDIX A

Second Order Runge-Kutta Derivation

For the function

$$\frac{dy}{dx} = F(X, y) \quad y = y_0 \text{ when } X = X_0$$

Taylor series expansion of $y_{n+1} = y(X_{n+1})$

$$y_{(n+1)} = y_n + F_n h + \left(\frac{\partial F_n}{\partial x} + \frac{\partial F_n}{\partial y} F_n \right) \frac{h^2}{2} + \dots \quad (1)$$

Where $F_n = F(X_n, y_n)$

$$\frac{\partial F_n}{\partial X} = \frac{\partial F(X_n, y_n)}{\partial X}$$

$$\frac{\partial F_n}{\partial y} = \frac{\partial F(X_n, Y_n)}{\partial y}$$

Assume approximation form

$$Y_{(n+1)} \doteq Y_n + \lambda_1 h F_n + \lambda_2 h F(X_n + \mu_1 h, Y_n + \mu_2 h F_n) \quad (2)$$

From Taylor series expansion

$$f(X+H, Y+K) = f(X, y) + H \frac{\partial f(x, y)}{\partial x} + K \frac{\partial f(x, y)}{\partial y} + \dots$$

For small values H and K

$$\begin{aligned} F(X_n + \mu_1 h, Y_n + \mu_2 h F_n) &= F(X_n, Y_n) + \mu_1 h \frac{\partial F(X_n, Y_n)}{\partial x} \\ &\quad + \mu_2 h \frac{\partial F(X_n, Y_n)}{\partial y} F_n + \dots \\ &= F_n + h \left[\mu_1 \frac{\partial F_n}{\partial x} + \mu_2 \frac{\partial F_n}{\partial y} F_n \right] + \dots \end{aligned}$$

Equation (2) becomes:

$$Y_{(n+1)} = Y_n + (\lambda_1 + \lambda_2) F_n h + \lambda_2 \left[\mu_1 \frac{\partial F_n}{\partial x} + \mu_2 \frac{\partial F_n}{\partial y} F_n \right] h^2 \quad (3)$$

Equations (1) and (3) are the same if

$$\lambda_1 = \lambda_2 = 1/2 \quad \text{and} \quad \mu_1 = \mu_2 = 1$$

Thus (2) becomes

$$Y_{(n+1)} = Y_n + \frac{h}{2} \left[F(X_n, Y_n) + F(X_n + h, Y_n + hF_n) \right]$$

Now for

$$K_1 = hF_n \quad K_2 = hF(X_n + h, Y_n + K_1)$$

$$Y_{(n+1)} = Y_n + 1/2(K_1 + K_2)$$

Now, for numerical solution of the state variable problem, we have given

$$\dot{X}_i = f_i(\bar{X}) \quad \text{and} \quad \bar{X}(nt)$$

Runge-Kutta says

$$X[(n+1)T] = X_{(nT)} + 1/2 (K_1 + K_2)$$

$$K_{i1} = Tf_i(\bar{X}_n)$$

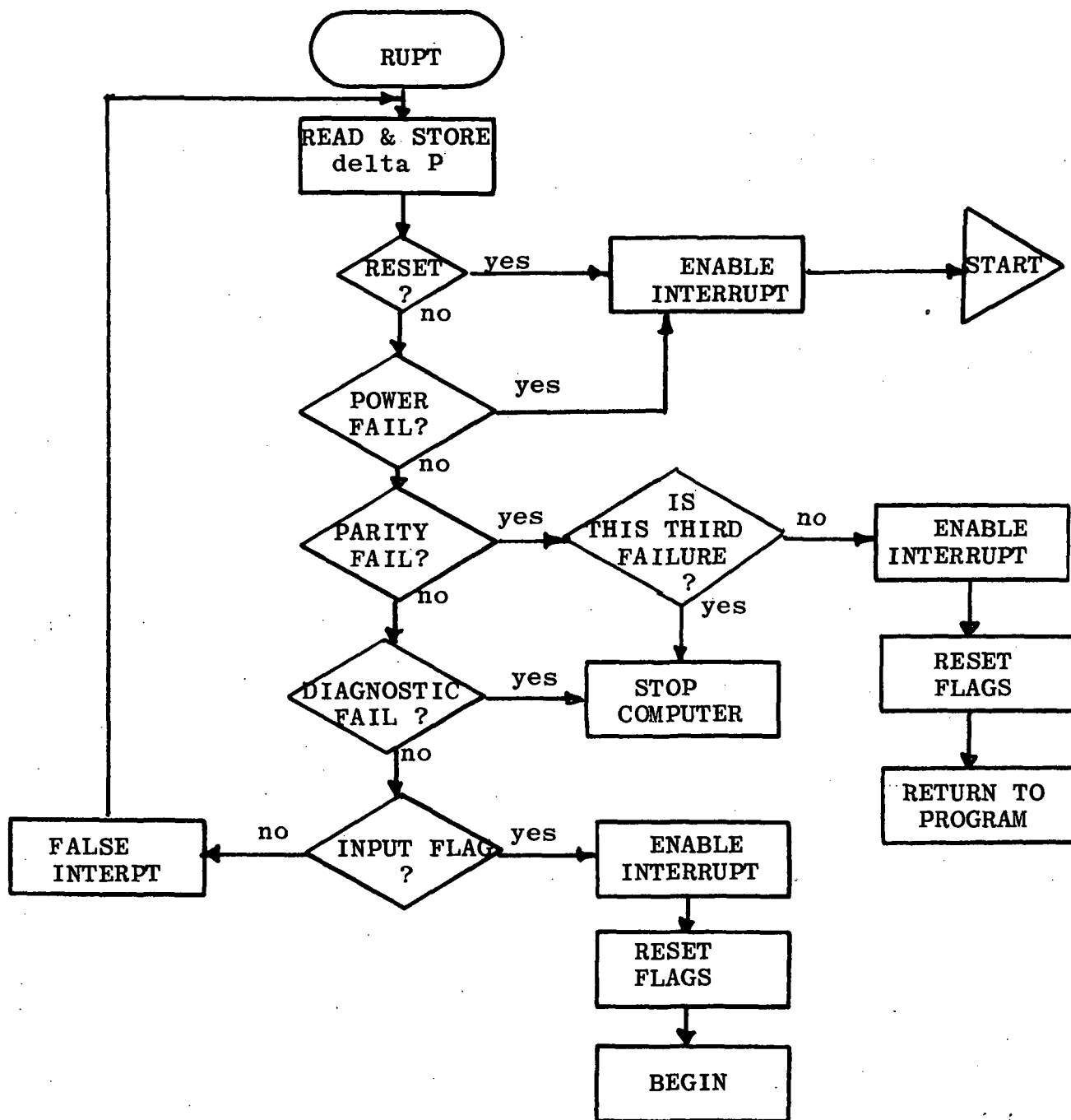
$$K_{i2} = Tf_i(\bar{X}_n + K_{i1})$$

$$= Tf_i \left[\bar{X}_n + Tf_i(\bar{X}_n) \right]$$

$$= Tf_i(\bar{X}_{n+1})$$

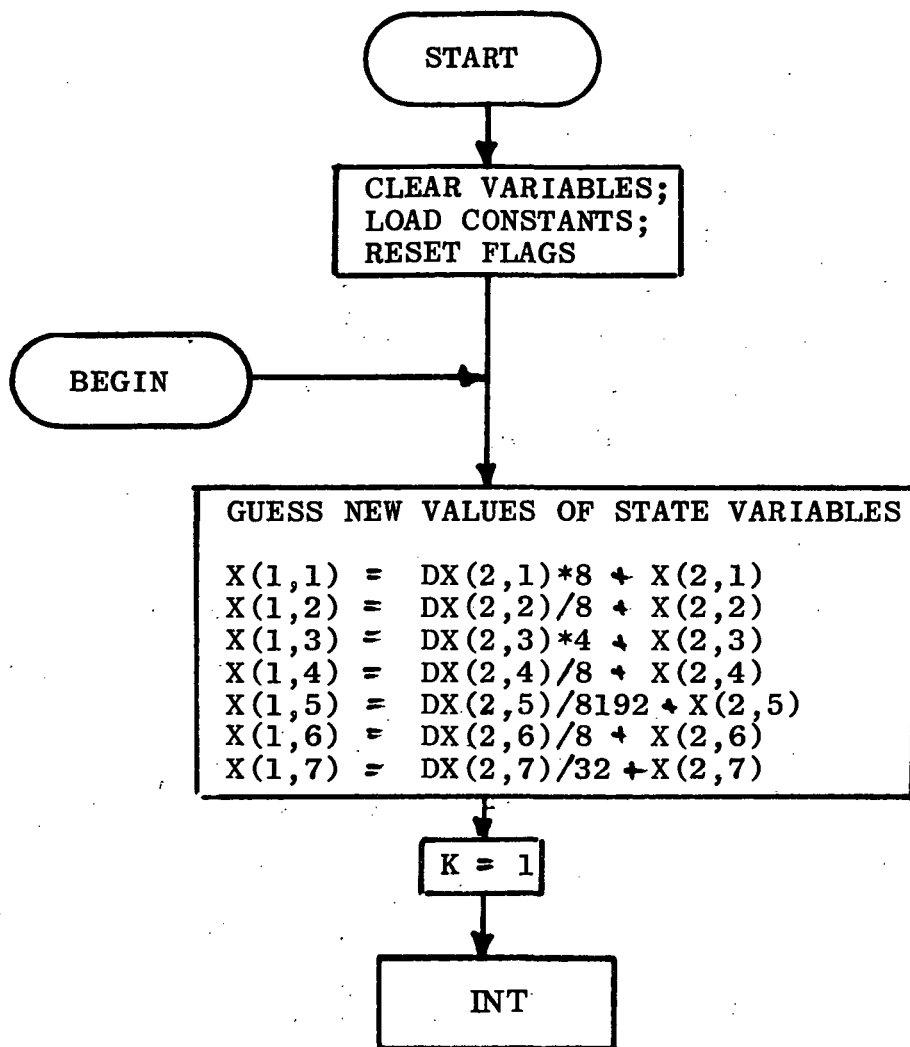
APPENDIX B

FLOW DIAGRAM OF COMPUTER PROGRAM



FLOW DIAGRAM OF INTERRUPT ROUTINE

FIGURE B-1

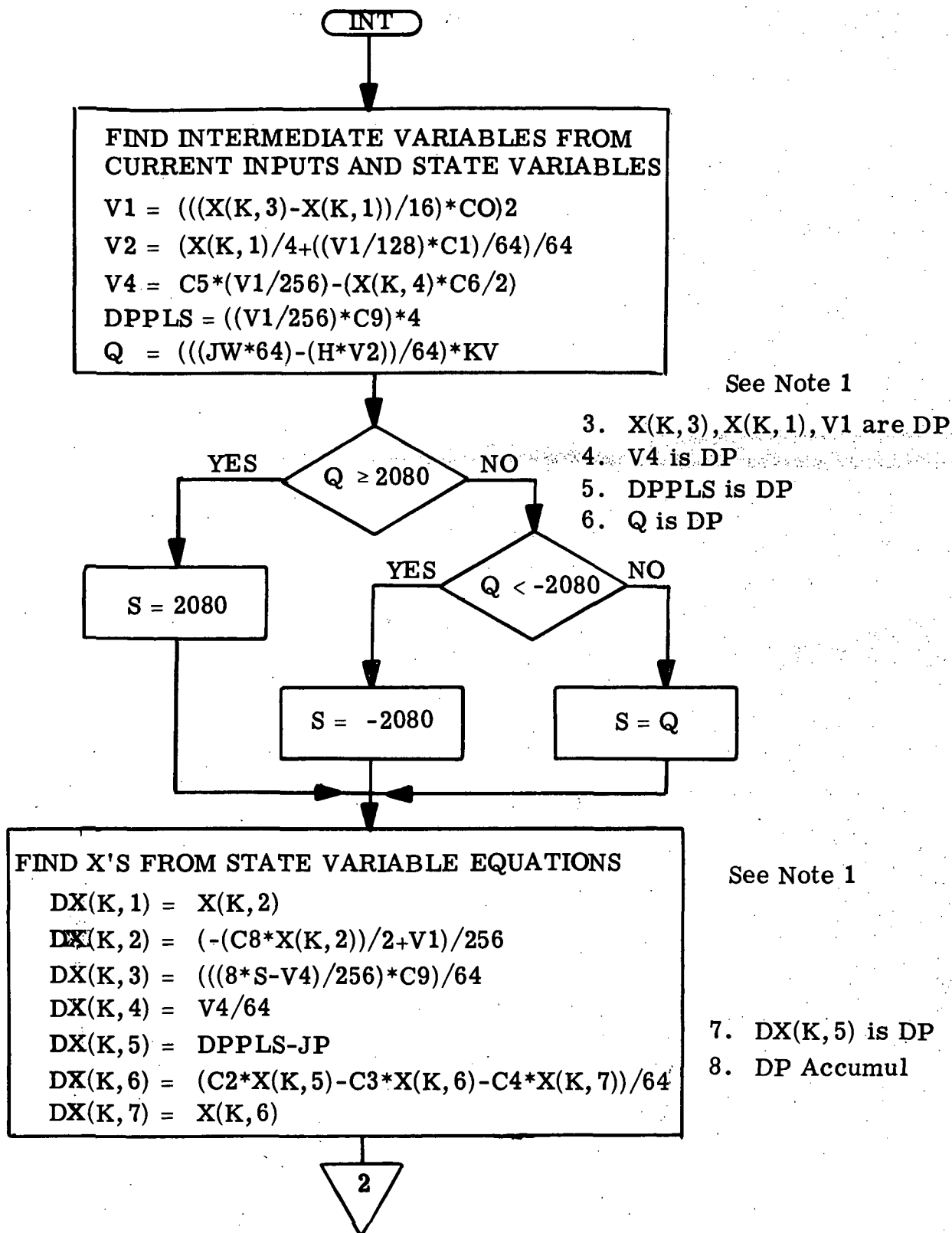


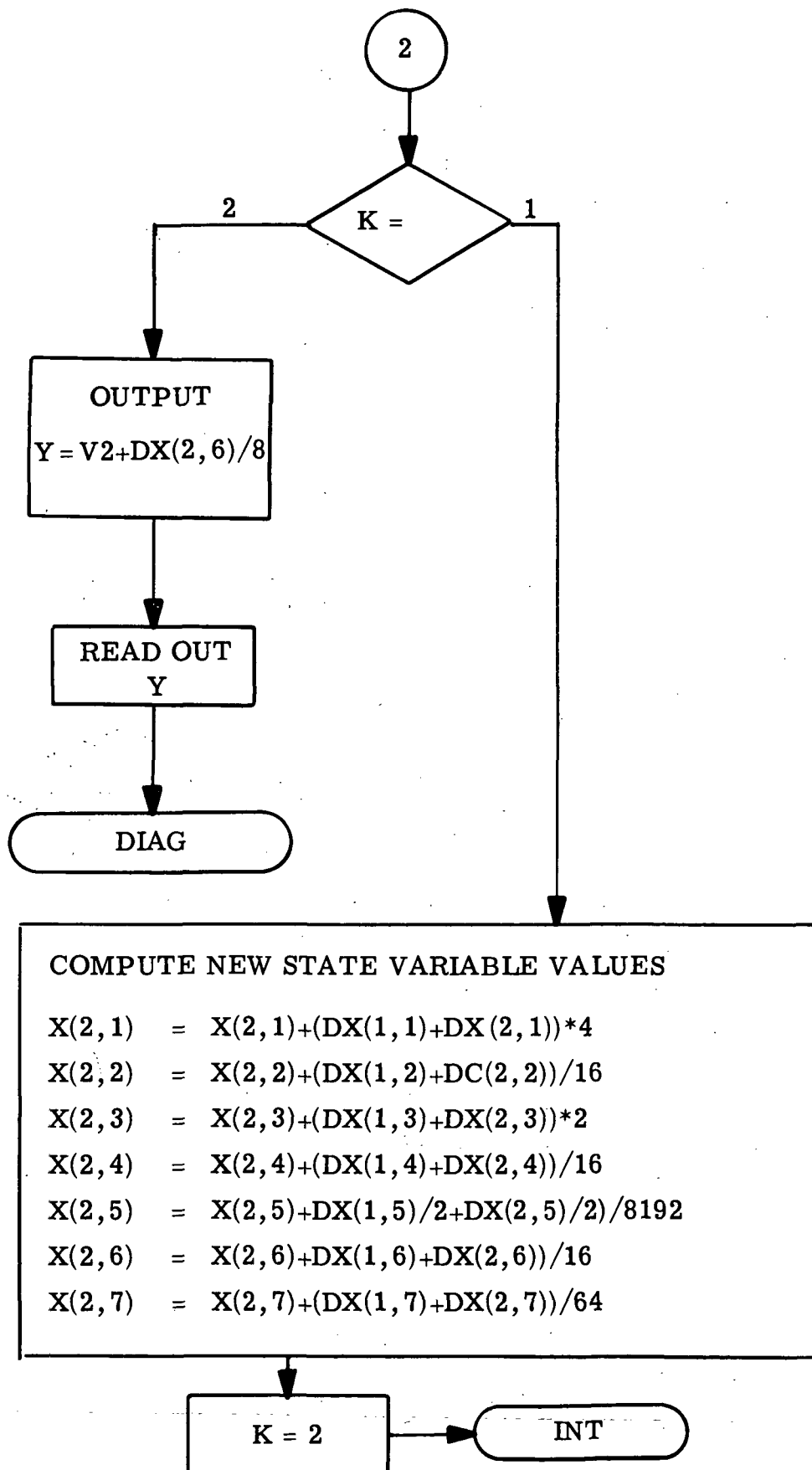
NOTE:

1. Implement all divisions as shift right with rounding
2. $X(1,1)$, $X(2,1)$, $X(1,3)$, $X(2,3)$ are double precision (DP)

MATH MODEL SOLUTION PROCESSING

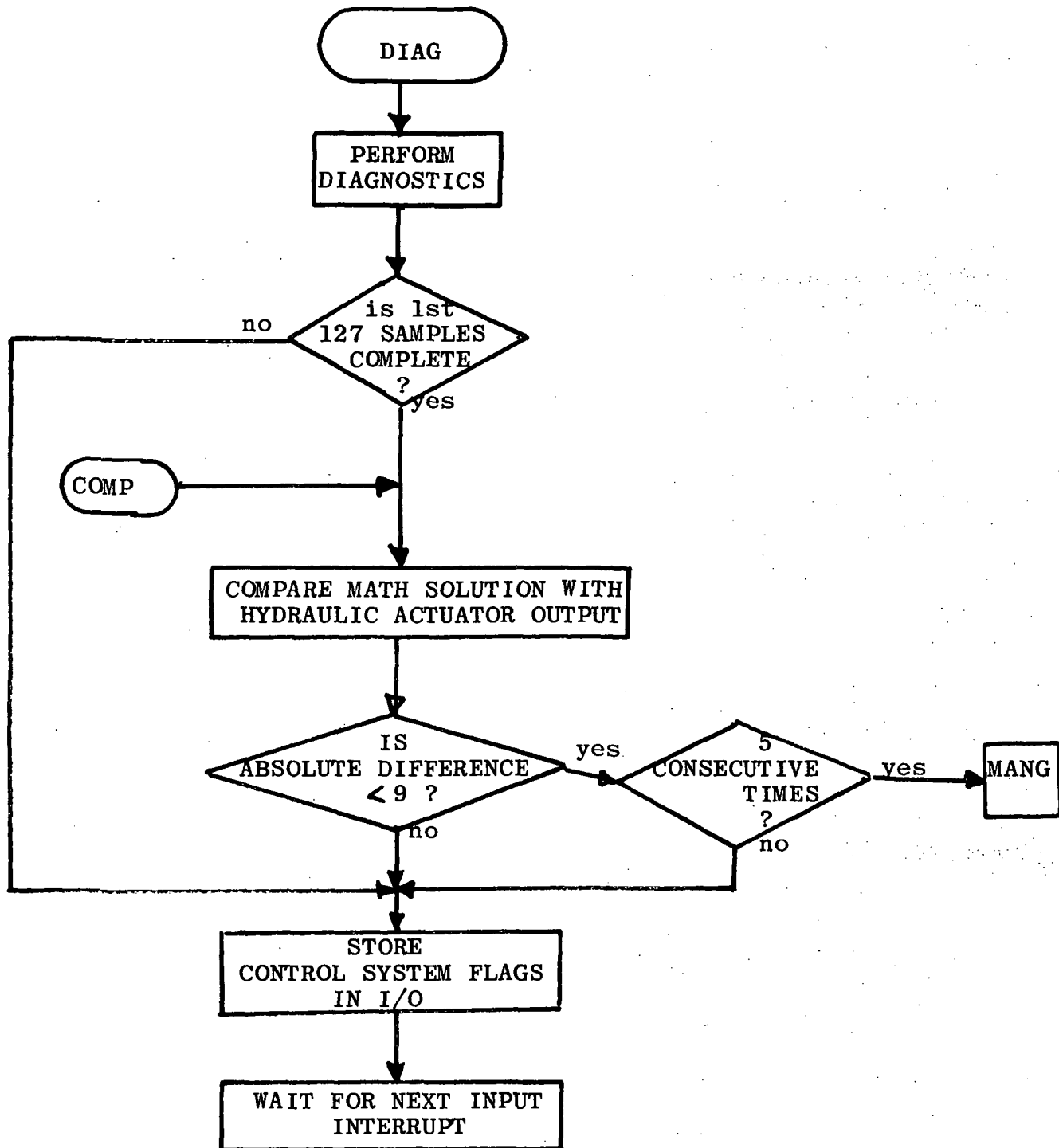
FIGURE B-2-1





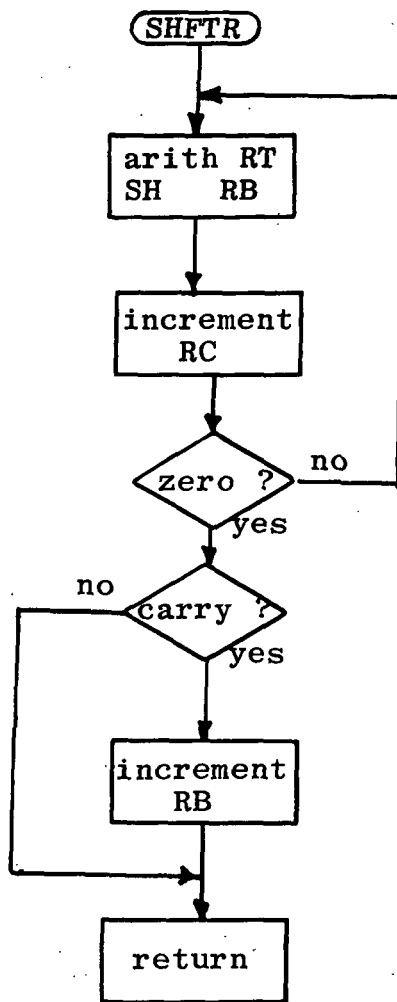
See Note 1

FIGURE B-2-3



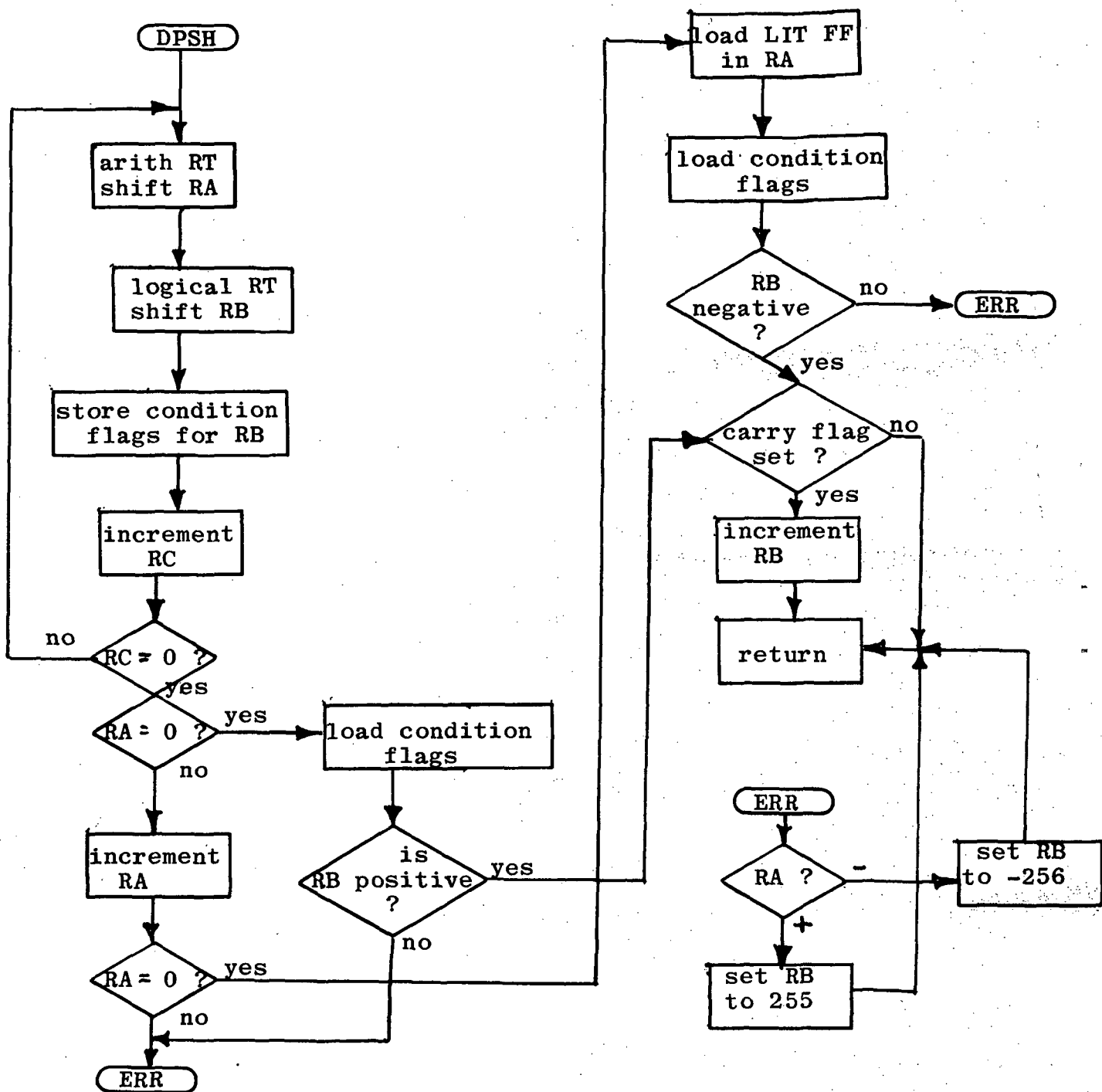
COMPARISON ROUTINE

FIGURE B-3



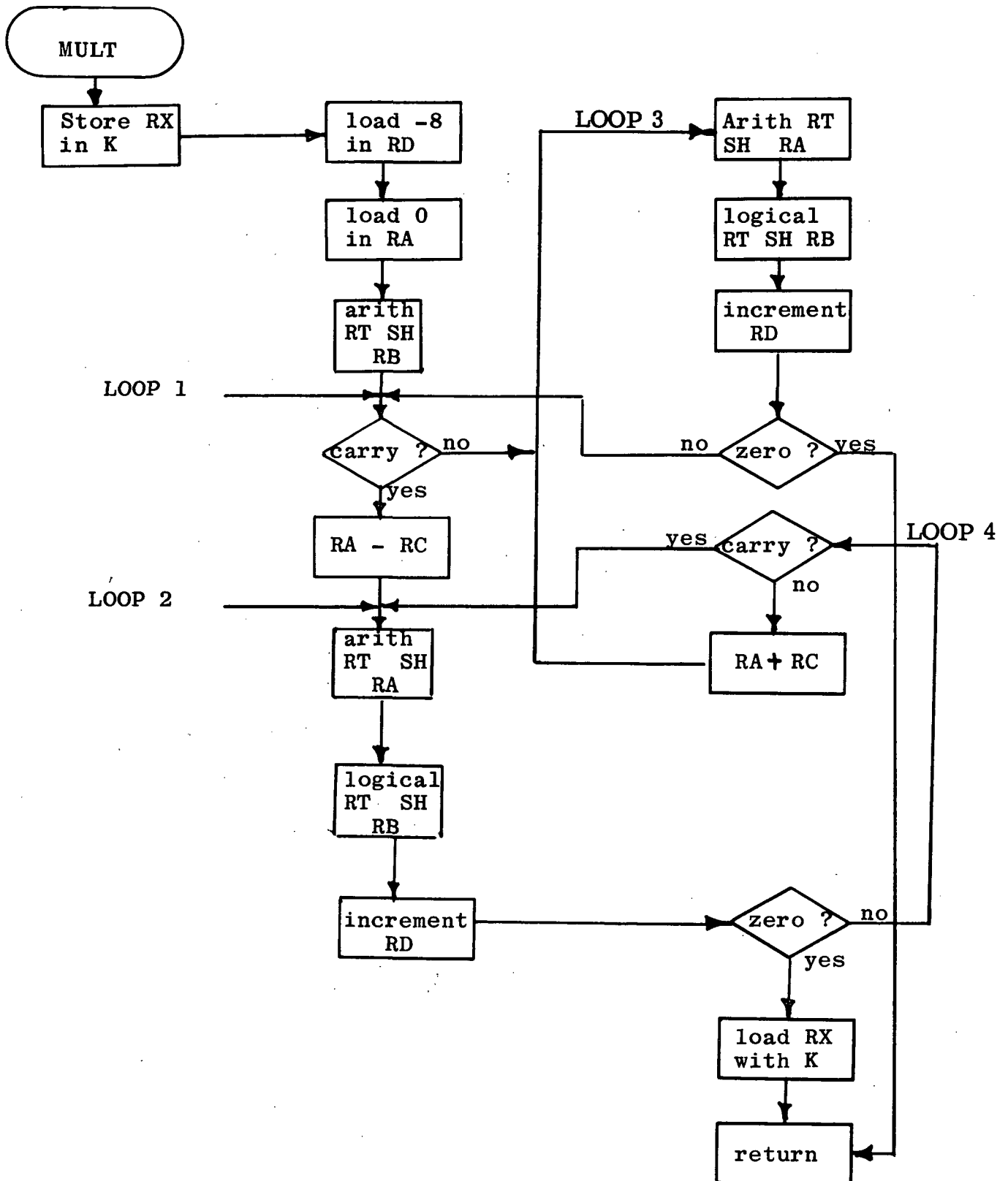
SHIFT RIGHT ROUTINE WITH ROUNDING

FIGURE B-4



DOUBLE PRECISION SHIFT RIGHT WITH ROUNDING

FIGURE B-5



MULTIPLY SUBROUTINE

FIGURE B-6

APPENDIX C

COMPUTER PROGRAM

LISTING

C1

P/L	CD REF	OP	OPERAND	MODIFIERS	COMMENTS OR SCI	P	D	CODE
000100	D RA	HEX	00	000 * A-REG		00	00	00
000200	D RB	HEX	00	001 * B-REG		00	01	00
000300	D RC	HEX	00	002 * C-REG		00	02	00
000400	D RX	HEX	00	003 * D-REG AND INDEX REG		00	03	00
000500	D RE	HEX	00	004 * EXTEND REG		00	04	00
000600	D RL	HEX	00	005 * LOCATION REG		00	05	00
000700	D R1	HEX	00	006 * PAGE REG 1		00	06	00
000800	D R1	HEX	00	007 * INTERRUPT REG		00	07	00
000900	D R2	HEX	00	008 * PAGE REG 2		00	08	00
001000	D SWSET	DS		009		00	09	00
001100	D VALID	EQU	SWSET			00	09	
	D STATUS	DS		00A		00	0A	00
001400	D WJ	HEX	00	00B		00	0B	00
	D CLEAR	DS		00C * MASK TO CLEAR FLAGS		00	0C	00
001600	D JPPL	HEX	00	00D		00	0D	00
001700	D YHAUL	HEX	00	00E		00	0E	00
	D FLAGS	DS		00F		00	0F	00
	D YMATH	DS		010 * MATH MODEL OUTPUT		00	10	00
	D DUMMY	DS				00	10	00
002000	D JW	HEX	00			00	11	00
002100	D Y	HEX	00			00	12	00
002130	D JPU	HEX	00			00	13	00
002160	D JPL	HEX	00			00	14	00
003800	D TEMPU	HEX	00			00	15	00

P/L	CO REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	R	D	CODE
003900	D	TEPL	HEX	00		00	16	00
004000	D	QU	HEX	00		00	17	00
004100	D	QL	HEX	00		00	18	00
004200	D	DPPLSU	HEX	00		00	19	00
004300	D	DPPLSL	HEX	00		00	1A	00
004400	D	V1U	HEX	00		00	1B	00
004500	D	V1L	HEX	00		00	1C	00
004600	D	V2	HEX	00		00	1D	00
004700	D	V4U	HEX	00		00	1E	00
004800	D	V4L	HEX	00		00	1F	00
004900	D	X1L	BSS	2		00	20	
005000	D	X1U	BSS	2		00	22	
005100	D	X2	BSS	2		00	24	
005200	D	X3L	BSS	2		00	26	
005300	D	X3U	BSS	2		00	28	
005400	D	X4	BSS	2		00	2A	
005500	D	X5	BSS	2		00	2C	
005600	D	X6	BSS	2		00	2E	
005700	D	X7	BSS	2		00	30	
005800	D	DX1	BSS	2		00	32	
005900	D	DX2	BSS	2		00	34	
006000	D	DX3	BSS	2		00	36	
006100	D	DX4	BSS	2		00	38	
006200	D	DX5L	BSS	2		00	3A	

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
006300	D	DX5U	BSS		2		00	3C	
006400	D	DX6	BSS		2		00	3E	
006500	D	DX7	BSS		2		01	00	
006700	D	SAMPLE DS					01	02	00
	D	SETSW DS					01	03	00
001300	D	DEABLE EQU	SETSW				01	03	
007200	D	PAGE	HEX		00		01	04	00
007300	D	TEST	DS				01	05	00
007500	D	K	DS				01	06	00
007600	D	PG	DS				01	07	00
007700	D	PL	DS				01	08	00
006600	D	TEMP	DS				01	09	00
	D	RCYCLE DS					01	0A	00
002300	D	CO	DEC		113		01	0B	71
002400	D	C1	DEC		126		01	0C	7E
002500	D	C2	DEC		79		01	0D	4F
002600	D	C3	DEC		74		01	0E	4A
002700	D	C4	DEC		86		01	0F	56
002800	D	C5	DEC		79		01	10	4F
002900	D	C6	DEC		75		01	11	4B
003000	D	C7	DEC		64		01	12	40
003100	D	C8	DEC		103		01	13	67
003200	D	C9	DEC		102		01	14	66
003300	D	KV	DEC		87		01	15	57

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR SCI	P	D	CODE
003400	D H		DEC		67		01	16	43
003500	D	QLIML	HEX	20		* QLM = 2080	01	17	20
003600	D	QLIMU	HEX	08		*	01	18	08
003700	D	ONE	HEX	01			01	19	01
006800	D	TOL	DS			* ALLOWABLE OUTPUT TOLERANCE	01	1A	00
006900	D	FIRST	DA	CLR			01	1B	24
007000	D	BEGAN	DA	BEGIN			01	1C	96
007100	D	PLACE	DA	MANG			01	1D	35
007400	D	TEST2	HEX	FD		* -3	01	1E	C7
	D	TP	DS				01	1F	00
	D	FAILS	DS				01	20	00
	D	YTEMP	DS				01	21	00
007800	D	LOC	EQU	RA			00	00	

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCJ	P	D	CODE
008000	C	*	ORG	000			00	00	A449
008100	C		ST	TEMP	RE		00	00	A449
	C		J	RUPT			00	01	E002
008400	C	RUPT	LL	00	RB		00	02	9100
	C		ST	JPU	RB		00	03	A113
	C		LD	JPPL	RA		00	04	800D
	C		ST	JPL	RA		00	05	A01A
	C		AND	RA	RA		00	06	0000
	C		BN/	HOP8			00	07	D90A
	C		LL	FF	RA		00	08	90FF
	C		ST	JPU	RA		00	09	A013
	C	HOP8	BI/	SKP		* RESTART ON RESET	00	0A	DD0D
	C		SLA	FLAGS	RA		00	0B	300F
	C		BC/	PAR			00	0C	DB10
	C	SKP	ST	CLEAR	RB		00	0D	A10C
	C		LL	29	RI		00	0E	8729 972
	C		LD	RI	RL		00	0F	8507
	C	PAR	SLA	RA	RA		00	10	3000
0055200	C		BC/	EIN			00	11	DB1A
0055300	C		INC	TEST2	RC		00	12	525E
0055400	C		ST	TEST2	RC		00	13	A25E
	C		BN	EAR			00	14	C916
	C		J	WATE			00	15	E2A4
	C	EAR	ST	CLEAR	RB		00	16	A10C
0055650	C		LD	PL	RI		00	17	8748

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
055700	C		LD	PG	RE		00	18	8447
055800	C		LD	RI	RL		00	19	8507
	C	EIN	SLA	RA	RA		00	1A	3000
	C		BC/	CONT1			00	18	D81D
	C		J	WATE			00	1C	E2A4
	C	CONT1	SLA	RA	RA		00	1D	3000
	C		BC/	RUPT			00	1E	D802
	C	CONT	ST	CLEAR	RB		00	1F	A10C
	C		LL		FD RB		00	20	91FD
	C		ST	TEST2	RB		00	21	A15E
	C		LD	BEGAN	RI		00	22	875C
056200	C		LD	RI	RL		00	23	8507
008330	C	CLR	ST	PG	RE		00	24	A447
008360	C		ST	PL	RL		00	25	A548
	C		LL		CC RX	* -52	00	26	93CC
008500	C		LL		00 RB		00	27	9100
008600	C	START	CNT	START	RX		00	28	8328
008700	C		ST	TEMP	RB,X		00	29	A949
	C		LL		10 RB		00	2A	9110
	C		ST	SAMPLE	RB		00	2B	A142
	C		LL		FB RB		00	2C	91FB
	C		ST	FAILS	RB		00	2D	A160
	C		LL		71 RB	* 113	00	2E	9171
	C		ST	CO	RB		00	2F	A148

P/L CD REF OP OPERAND MODIFIERS COMMENTS OR BCI

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
C			LL	7E RB	* 126		00	30	917E
C			ST	C1	RB		00	31	A14C
C			LL	4F RB	* 79		00	32	914F
C			ST	C2	RB		00	33	A14D
C			ST	C5	RB		00	34	A150
C			LL	4A RB	* 74		00	35	914A
C			ST	C3	RB		00	36	A14E
C			LL	56 RB	* 86		00	37	9156
C			ST	C4	RB		00	38	A14F
C			LL	4B RB	* 75		00	39	914B
C			ST	C6	RB		00	3A	A151
C			LL	40 RB	* 64		00	3B	9140
C			ST	C7	RB		00	3C	A152
C			LL	67 RB	* 103		00	3D	9167
C			ST	C8	RB		00	3E	A153
C			LL	66 RB	* 103		00	3F	9166
C			ST	C9	RB		00	40	A154
C			LL	57 RB	* 87		00	41	9157
C			ST	KV	RB		00	42	A155
C			LL	43 RB	* 67		00	43	9143
C			ST	H	RB		00	44	A156
C			LL	20 RB			00	45	9120
C			ST	GLIML	RB		00	46	A157
C			LL	08 RB			00	47	9108

P/L	CD REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
	C	ST	OLIMU	RB		00	48	A158
	C	LL	01 RB			00	49	9101
	C	ST	ONE	RB		00	4A	A159
	C	LL	²⁴ 22 RB			00	4B	9123 9124
	* E 000 COL 7 BAD							
	* E 700 BAD LITERAL							
	C	ST	FIRST	RB		00	4C	8000 A15
	C	LL	⁹⁶ 8A RB			00	4D	912A 919
	C	ST	BEGAN	RB		00	4E	A18C
	C	LL	³⁵ 8A RB			00	4F	9101 913
	C	ST	PLACE	RB		00	50	A15D
	C	LL	FD RB		* -3	00	51	91FD
	C	ST	TEST2	RB		00	52	A15E
	C	LL	08 RB			00	53	9108
	C	ST	STATUS	RB	* SET FLAG IND COMPUTER GOOD	00	54	A10A
	C	LL	00 RB			00	55	9100
	C	ST	SETSW	RB		00	56	A143
	C	ST	DUMMY	RB		00	57	A110
	C	LL	09 RB			00	58	9109
	C	ST	TOL	RB		00	59	A15A
009000	C	J	BEGIN			00	5A	E096
009100	C	MULT	ST K	RX	* MULTIPLY SUBROUTINE - BOOTH 1	00	5B	A346
009200	C	LL	F8 RX		* LD -8 IN RX	00	5C	93F8
	C	LL	OF RA			00	5D	900F
	C	AND	RE	RA		00	5E	0004

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
	C		ST	RA	RE		00	5F	A400
009600	C		LL		00 RA	1	00	60	9000
009700	C		SRL	RB	RB	1	00	61	3001
	C		ST	TP	RE		00	62	A45F
	C	LOOP1	LD	TP	RE		00	63	845F
	C	BC/ LOOP3					00	64	D86E
009900	C		SUB	RC	RA	1	00	65	7002
010000	C	LOOP2	SRA	RA	RA	1	00	66	3400
010100	C		SRL	RB	RB	1	00	67	3001
	C		ST	TP	RE		00	68	A45F
010200	C		INC	RX	RX	1	00	69	5303
010300	C		BZ/	LOOP4		1	00	6A	DA74
010400	C	RETURN	LD	K	RX	1	00	6B	8346
010500	C		LD	TEMP	RE	1	00	6C	8449
010600	C		LD	RI	RL	1	00	6D	8507
010700	C	LOOP3	SRA	RA	RA	1	00	6E	3400
010800	C		SRL	RB	RB	1	00	6F	3001
	C		ST	TP	RE		00	70	A45F
010900	C		INC	RX	RX	1	00	71	5303
011000	C		BZ/	LOOP1		1	00	72	DA63
011100	C		J	RETURN		1	00	73	E06B
	C	LOOP4	LD	TP	RE		00	74	845F
	C		BC	LOOP2			00	75	C866
011300	C		ADD	RC	RA	1	00	76	4002

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
011400	C		J	LOOP3		1	00	77	E06E
011500	C	DPSH	SRA	RA	RA	* DP SHIFT RIGHT SUBROUTINE	00	78	3400
	C		SRL	RB	RB		00	79	3D01
	C		ST	TP	RE		00	7A	A45F
	C		CNT	DPSH	RC		00	7B	B278
	C		AND	RA	RA		00	7C	0000
	C		BZ	PLUS			00	7D	CA84
	C		INC	RA	RA		00	7E	5000
	C		BZ/	ERR			00	7F	DABA
	C		LL		FF RA		00	80	90FF
	C		LD	TP	RE		00	81	845F
	C		BN/	ERR			00	82	D98A
	C		J	CCC			00	83	E086
	C	PLUS	LD	TP	RE		00	84	845F
	C		BN	ERR			00	85	C98A
	C	CCC	BC/	EXIT3			00	86	D888
	C		INC	RB	RB		00	87	5101
	C	EXIT3	LD	TEMP	RE		00	88	8449
	C		LD	RI	RL		00	89	8507
	C	ERR	AND	RA	RA		00	8A	0000
	C		BN	NEG1			00	8B	C98E
	C		LL		7F RB		00	8C	917F
	C		J	EXIT3			00	8D	E088
	C	NEG1	LL		80 RB		00	8E	9180

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
	C		J	EXIT3			00	8F	E088
013000	C	SHFTR	CNT	SHFTR	RC	* SHIFT RIGHT SUBROUTINE 2	00	90	B290
013100	C	SRA	RB	RB	RB	2	00	91	3501
	C	BC/	EXIT2				00	92	DB94
	C	INC	RB	RB	RB		00	93	5101
013700	C	EXIT2	LD	TEMP	RE	2	00	94	8A48
013800	C	LD	RI	RL	RL	2	00	95	8507
013830	C	BEGIN	ST	PG	RE		00	96	A447
013860	C	ST	PL	PL	RL		00	97	A548
013900	C	LL	LL	00 RA		* GUESS NEW VALUES OF STATE VAR	00	98	9000
014000	C	LL	LL	01 RX			00	99	9301
014400	C	LD	DX1	RB,X			00	9A	8932
	C	LL	LL	00 RA			00	9B	9000
	C	AND	RB	RB			00	9C	0101
	C	BN/	HOP5				00	9D	D99F
	C	LL	LL	FF RA			00	9E	90FF
	C	HOP5	SLA	RB	RB		00	9F	3101
014600	C	SLL	RA	RA			00	A0	3800
014700	C	SLA	RB	RB			00	A1	3101
014800	C	SLL	RA	RA			00	A2	3800
014900	C	SLA	RB	RB			00	A3	3101
015000	C	SLL	RA	RA			00	A4	3800
015100	C	ADD	X1L	RB,X			00	A5	4920
015200	C	ADD	X1U	RA,X,MP			00	A6	4C22

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
015300	C		ST	X1U	RA		00	A7	A022
015400	C		ST	X1L	RB	* X(1,1)=DX(2,1)+8+X(2,1)	00	A8	A120
015500	C		LD	DX2	RB,X		00	A9	8934
015600	C		LL		FD RC	* -3	00	AA	92FD
015700	C		ST	TEMP	RE		00	AB	A449
015800	C		LK	SHFTR	RI		00	AC	F790
015900	C		ADD	X2	RB,X		00	AD	4924
016000	C		ST	X2	RB	* X(1,2)=DX(2,2)/8+X(2,2)	00	AE	A124
016300	C		LL		00 RA		00	AF	9000
016100	C		LD	DX3	RB,X		00	B0	8936
	C		AND	RB	RB		00	B1	0101
	C		BN/	HOP6			00	B2	D984
	C		LL		FF RA		00	B3	90FF
	C	HOP6	SLA	RB	RB		00	B4	3101
016400	C		SLL	RA	RA		00	B5	3800
016500	C		SLA	RB	RB		00	B6	3101
016600	C		SLL	RA	RA		00	B7	3800
016700	C		ADD	X3L	RB,X		00	B8	4926
016800	C		ADD	X3U	RA,X,MP		00	B9	4C28
016900	C		ST	X3U	RA		00	BA	A028
017000	C		ST	X3L	RB	* X(1,3)=DX(2,3)+4+X(2,3)	00	BB	A126
017100	C		LD	DX4	RB,X		00	BC	8938
017200	C		LL		FD RC	* -3	00	BD	92FD
017300	C		ST	TEMP	RE		00	BE	A449

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
017400	C		LK	SHFTR	RI		00	BF	F790
017500	C		ADD	X4	RB,X		00	C0	492A
017600	C		ST	X4	RB	* X(1,4)=DX(2,4)/8+X(2,4)	00	C1	A12A
017700	C		LD	DX5U	RB,X		00	C2	893C
017800	C		LL		FB RC	* -5	00	C3	92FB
017900	C		ST	TEMP	RE		00	C4	A449
018000	C		LK	SHFTR	RI		00	C5	F790
018100	C		ADD	X5	RB,X		00	C6	492C
018200	C		ST	X5	RB	* X(1,5)=DX(2,5)/8192+X(2,5)	00	C7	A12C
018300	C		LD	DX6	RB,X		00	C8	893E
018400	C		LL		FD RC	* -3	00	C9	92FD
018500	C		ST	TEMP	RE		00	CA	A449
018600	C		LK	SHFTR	RI		00	CB	F790
018700	C		ADD	X6	RB,X		00	CC	492E
018800	C		ST	X6	RB	* X(1,6)=DX(2,6)/8+X(2,6)	00	CD	A12E
018900	C		LD	DX7	RB,X		00	CE	8940
019000	C		LL		FB RC	* -5	00	CF	92FB
019100	C		ST	TEMP	RE		00	D0	A449
019200	C		LK	SHFTR	RI		00	D1	F790
019300	C		ADD	X7	RB,X		00	D2	4930
019400	C		ST	X7	RB	* X(1,7)=DX(2,7)/32+X(2,7)	00	D3	A130
019500	C		LL		00 RX	* CLEAR INDEX REGISTER	00	D4	9300
019530	C	INT	ST	PG	RE		00	D5	A447
019560	C		ST	PL	RL		00	D6	A548

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR ACI	P	D	CODE
019600	C		LD	X3L	RB,X	* V1=((X(K,3)-X(K,1))/16)*C0)*2	00	D7	8926
019700	C		LD	X3U	RA,X		00	D8	8828
019800	C		SUB	X1L	RB,X		00	D9	7920
019900	C		SUB	X1U	RA,X,MP		00	DA	7C22
020100	C		LL		FC RC	* -4	00	DB	92FC
020200	C		ST	TEMP	RE		00	DC	A449
020300	C		LK	DPSH	RI		00	DD	F778
020400	C		LD	C0	RC		00	DE	8248
020500	C		LK	MULT	RI		00	DF	F75B
020600	C		SLA	RB	RB		00	E0	3101
020700	C		SLL	RA	RA		00	E1	3800
020800	C		ST	V1L	RB		00	E2	A11C
020900	C		ST	V1U	RA		00	E3	A01B
021000	C		SLA	V1L	RB	* V2=(X(K,1)/4+((V1/256)*C1)	00	E4	311C
021100	C		INC	V1U	RB,MP	* /32)/64	00	E5	551B
021200	C		LD	C1	RC		00	E6	824C
021300	C		ST	TEMP	RE		00	E7	A449
021400	C		LK	MULT	RI		00	E8	F75B
021500	C		LL		FB RC		00	E9	92F8
021600	C		REPEAT	SRA RA	RA		00	EA	3400
021700	C		SRL	RB	RB		00	EB	3D01
021800	C		INC	RC	RC		00	EC	5202
021900	C		BZ/	REPEAT			00	ED	DAEA
022000	C		ST	TEMPU	RA		00	EE	A015

COMMENTS OR BCI

P D CODE

P/L CD REF OP OPERAND MODIFIERS

C		C	ST	TEMPL	RB		00 EF	A116
C		LD	X1U	RA,X			00 F0	8822
C		LD	X1L	RB,X			00 F1	8920
C		SRA	RA	RA			00 F2	3400
C		SRL	RB	RB			00 F3	3D01
C		SRA	RA	RA			00 F4	3400
C		SRL	RB	RB			00 F5	3D01
C		ADD	TEMPL	RB			00 F6	4116
C		ADD	TEMPU	RA,MP			00 F7	4418
C		LL		FA RC			00 F8	92FA
C		ST	TEMP	RE			00 F9	A449
C		LK	DPSH	RI			00 FA	F778
C		ST	V2	RB			00 FB	A11D
022700	C	LD	X4	RB,X		* V4=C5*(V1/256)=(X(K,4)+C6)/2)	00 FC	892A
022800	C	LD	C6	RC			00 FD	8251
C		J	PAGE1				00 FE	E100

P/L	CD	REF	OP	ORG	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
022900	C	C	PAGE1	ST	TEMP	100 RE		01	00	A449
023000	C	C	LL	LL	00 RE	RI		01	01	9400
023100	C	C	LK	LK	MULT	RI		01	02	F75B
023200	C	C	NOP					01	03	8000
023300	C	C	SRA	RA	RA			01	04	3400
023400	C	C	SRL	RB	RB			01	05	3001
023500	C	C	INC	RB	RB,MP			01	06	5501
023600	C	C	INC	RA	RA,MP			01	07	5400
023700	C	C	ST	TEMPU	RA			01	08	A01B
023800	C	C	ST	TEMPL	RB			01	09	A116
023900	C	C	SLA	V1L	RB			01	0A	311C
024000	C	C	INC	V1U	RB,MP			01	0B	551B
024100	C	C	LD	C5	RC			01	0C	8250
024200	C	C	ST	TEMP	RE			01	0D	A449
024300	C	C	LL	LL	00 RE			01	0E	9400
024400	C	C	LK	MULT	RI			01	0F	F75B
024500	C	C	SUB	TEMPL	RB			01	10	7116
024600	C	C	SUB	TEMPU	RA,MP			01	11	7415
024700	C	C	ST	V4U	RA			01	12	A01E
024800	C	C	ST	V4L	RB			01	13	A11F
024900	C	C	SLA	V1L	RB			01	14	311C
025000	C	C	LL	LL	00 RB			01	15	9100
025100	C	C	ADD	V1U	RB,MP			01	16	451B
025200	C	C	LD	C9	RC			01	17	8254

* ((V1/256)*C9)*4

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
024500	C		ST	TEMP	RE		01	18	A449
	C		LL		00 RE		01	19	9400
024600	C		LK	MULT	RI		01	1A	F758
024700	C		SLA	RB	RB		01	18	3101
024800	C		SLL	RA	RA		01	1C	3800
024900	C		SLA	RB	RB		01	1D	3101
025000	C		SLL	RA	RA		01	1E	3800
025100	C		ST	DPPLSU	RA		01	1F	A019
025200	C		ST	DPPLSL	RB		01	20	A11A
025230	C	QCAL	ST	PG	RE		01	21	A447
025260	C		ST	PL	RL		01	22	A548
025300	C		LD	V2	RB	* Q=((JW*64-H*V2)/64)*KV	01	23	B11D
025400	C		LD	H	RC		01	24	B256
025500	C		ST	TEMP	RE		01	25	A449
	C		LL		00 RE		01	26	9400
025600	C		LK	MULT	RI		01	27	F758
025700	C		ST	TEMPU	RA		01	28	A015
025800	C		ST	TEMPL	RB		01	29	A11A
	C		LD	WJ	RB		01	2A	8108
	C		LL		05 RC	* MULT INPUT BY 5/0	01	28	9205
	C		ST	TEMP	RE		01	2C	A449
	C		LL		00 RE		01	2D	9400
	C		LK	MULT	RI		01	2E	F758
	C		LL		FD RC		01	2F	92FD

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
	C		ST	TEMP	RE		01	30	A449
	C		LL		00 RE		01	31	9400
	C		LK	DPSH	RI		01	32	F778
	C		ST	JW	RB		01	33	A111
	C		LL		00 RA		01	34	9000
	C		AND	RB	RB		01	35	0101
	C		BN/	HOP7			01	36	D938
	C		LL		FF RA		01	37	90FF
	C	HOP7	LL		FA RC	* -6	01	38	92FA
026200	C	LP5	SLA	RB	RB		01	39	3101
	C		SLL	RA	RA		01	3A	3800
	C		CNT	LP5	RC		01	3B	B239
	C		NOP				01	3C	8000
026500	C		SUB	TEMPL	RB		01	3D	7116
026600	C		SUB	TEMPU	RA,MP		01	3E	7415
026700	C		LL		FA RC	* -6	01	3F	92FA
026800	C		ST	TEMP	RE		01	40	A449
	C		LL		00 RE		01	41	9400
026900	C		LK	DPSH	RI		01	42	F778
027000	C		LD	KV	RC		01	43	8255
	C		LL		00 RE		01	44	9400
027100	C		LK	MULT	RI		01	45	F75B
027200	C		ST	QU	RA		01	46	A017
027300	C		ST	QL	RB		01	47	A118

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
027400	C		AND	RA	RA	* LIMIT 0 TO +2080 OR -2080	01	48	0000
027500	C		BN	GNEG			01	49	C952
027600	C		SUB	QLIML	RB		01	4A	7157
027700	C		SUB	QLIMU	RA,MP		01	4B	7458
027800	C		BN	CONTIN			01	4C	C959
027900	C		LD	QLIMU	RA		01	4D	8058
028000	C		LD	QLIML	RB		01	4E	8157
028100	C		ST	QU	RA		01	4F	A017
028200	C		ST	QL	RB		01	50	A118
028300	C		J	CONTIN			01	51	E159
028400	C	GNEG	ADD	QLIML	RB		01	52	4157
028500	C		ADD	QLIMU	RA,MP		01	53	4458
	C		BP	CONTIN			01	54	C859
028700	C		TC	QLIML	RB		01	55	6157
028800	C		TC	QLIMU	RA,MP		01	56	6458
028900	C		ST	QU	RA		01	57	A017
029000	C		ST	QL	RB		01	58	A118
029030	C	CONTIN	ST	PG	RE	* X DOT CALCULATIONS	01	59	A447
029060	C		ST	PL	RL		01	5A	A548
029100	C		LD	X2	RB,X	* DX(K,1)=X(K,2)	01	5B	8924
029200	C		ST	DX1	RB,X		01	5C	A932
029300	C		TC	C8	RC	* DX(K,2) =	01	5D	6253
029400	C		ST	TEMP	RE	* (=(C8*X(K,2))/2 V1)/256	01	5E	A449
	C		LL		00 RE		01	5F	9400

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCJ	P	D	CODE
029500	C		LK	MULT	RI		01	60	F75B
029600	C		SRA	RA	RA		01	61	3400
	C		SRL	RB	RB		01	62	3001
	C		INC	RB	RB,MP		01	63	5501
	C		INC	RA	RA,MP		01	64	5400
029800	C		ADD	V1L	RB		01	65	411C
029900	C		ADD	V1U	RA,MP		01	66	441B
	C		SLA	RB	RB		01	67	3101
	C		INC	RA	RA,MP		01	68	5400
	C		ST	DX2	RA,X		01	69	A834
	C		SLA	QL	RB	* DX(K,3) =	01	6A	311B
030200	C		SLL	QU	RA	* ((18*S-V4)/256)*C9)/64	01	6B	3817
030300	C		SLA	RB	RB		01	6C	3101
030400	C		SLL	RA	RA		01	6D	3800
030500	C		SLA	RB	RB		01	6E	3101
030600	C		SLL	RA	RA		01	6F	3800
030700	C		SUB	V4L	RB		01	70	711F
030800	C		SUB	V4U	RA,MP		01	71	741E
	C		SLA	RB	RB		01	72	3101
	C		INC	RA	RB,MP		01	73	5500
031000	C		LD	C9	RC		01	74	8254
031100	C		ST	TEMP	RE		01	75	A449
	C		LL		00 RE		01	76	9400
031200	C		LK	MULT	RI		01	77	F75B

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
031300	C		LL		FA RC	* -6	01	78	92FA
	C		ST	TEMP	RE		01	79	A449
	C		LL		00 RE		01	7A	9400
031400	C		LK	DPSH	RI		01	7B	F778
031500	C		ST	DX3	RB,X		01	7C	A936
031600	C		LD	V4U	RA	* DX(K,4) = V4/64	01	7D	801E
031700	C		LD	V4L	RB		01	7E	811F
031800	C		LL		FA RC	* -6	01	7F	92FA
031900	C		ST	TEMP	RE		01	80	A449
	C		LL		00 RE		01	81	9400
032000	C		LK	DPSH	RI		01	82	F778
032100	C		ST	DX4	RB,X		01	83	A938
032200	C		LD	DPPLSJ	RA	* DX(K,5) = DPPLS=JP	01	84	8019
032300	C		LD	DPPLSL	RB		01	85	811A
032400	C		SUB	JPL	RB		01	86	7114
032500	C		SUB	JPU	RA,MP		01	87	7413
032600	C		ST	DX5U	RA		01	88	A03C
032700	C		ST	DX5L	RB		01	89	A13A
032720	C		LL		00 RA		01	8A	9000
032740	C		ST	DX5L	RA		01	8B	A03A
032760	C		ST	DX5U	RA		01	8C	A03C
032800	C		LD	C2	RB	* DX(K,6) = (C2*X(K,5)	01	8D	814D
032900	C		LD	X5	RC,X	-C3*X(K,6)=C4*X(K,7))/64	01	8E	8A2C
033000	C		ST	TEMP	RE		01	8F	A449

COMMENTS OR BCI

P/L	CD	REF	OP	OPERAND	MODIFIERS		P	D	CODE
	C		LL		00 RE		01	90	9400
033100	C		LK	MULT	RI		01	91	F75B
033200	C		ST	TEMPU	RA		01	92	A015
033300	C		ST	TEMPL	RB		01	93	A116
033600	C		LD	X6	RB,X		01	94	892E
033700	C		TC	C3	RC		01	95	624E
033800	C		ST	TEMP	RE		01	96	A449
033900	C		LL		00 RE		01	97	9400
034000	C		LK	MULT	RI		01	98	F75B
034100	C		ADD	TEMPL	RB		01	99	4116
034200	C		ADD	TEMPU	RA,MP		01	9A	4415
034300	C		ST	TEMPU	RA		01	9B	A015
034400	C		ST	TEMPL	RB		01	9C	A116
034500	C		LD	X7	RB,X		01	9D	8930
034600	C		TC	C4	RC		01	9E	624F
034700	C		ST	TEMP	RE		01	9F	A449
034800	C		LL		00 RE		01	A0	9400
034900	C		LK	MULT	RI		01	A1	F75B
035000	C		ADD	TEMPL	RB		01	A2	4116
035100	C		ADD	TEMPU	RA,MP		01	A3	4415
035200	C		LL		FA RC	* =6	01	A4	92FA
035300	C		ST	TEMP	RE		01	A5	A449
035400	C		LL		00 RE		01	A6	9400
035500	C		LK	DPSH	RI		01	A7	F778

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
035600	C		ST	DX6	RB,X		01	A8	A93E
035700	C		LD	X6	RB,X		01	A9	B92E
035800	C		ST	DX7	RB,X	* DX(K,7)=X(K,6)	01	AA	A940
035900	C		AND	RX	RX		01	AB	0303
	C		BZ	PUT			01	AC	CAAE
	C		J	OUTPUT			01	AD	E223
	C	PUT	INC	RX	RX		01	AE	5303
036130	C	VAR1	ST	PG	RE		01	AF	A447
036160	C		ST	PL	RL		01	B0	A548
036200	C		LL		OF RA	* CLEAR FLAGS	01	B1	900F
036300	C		AND	RE	RA		01	B2	0004
036400	C		ST	RE	RA		01	B3	A004
	C		LL		00 RA		01	B4	9000
	C		LL		00 RC		01	B5	9200
	C		LD	DX1	RB,X		01	B6	B932
	C		AND	RB	RB		01	B7	0101
	C		BN/	HOP1			01	B8	D98A
	C		LL		FF RA		01	B9	90FF
	C	HOP1	ST	TP	RB		01	BA	A15F
	C		LD	DX1	RB		01	BB	B132
	C		AND	RB	RB		01	BC	0101
	C		BN/	HOP2			01	BD	D98F
	C		LL		FF RC		01	BE	92FF
	C	HOP2	ADD	TP	RB		01	BF	A15F

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR RCI	P	D	CODE
	C		ADD	RC	RA,MP		01	C0	4402
036900	C		SLA	RB	RB		01	C1	3101
037000	C		SLL	RA	RA		01	C2	3800
037100	C		SLA	RB	RB		01	C3	3101
037200	C		SLL	RA	RA		01	C4	3800
037300	C		ADD	X1L	RB,X		01	C5	4920
037400	C		ADD	X1U	RA,X,MP		01	C6	4C22
037500	C		ST	X1U	RA,X * X(2,1)=X(2,1)+		01	C7	A822
	C		ST	X1L	RB,X * (DX(1,1)+DX(2,1)))+4		01	C8	A920
037630	C		ST	PG	RE		01	C9	A447
037660	C		ST	PL	RL		01	CA	A548
037700	C		SRA	DX2	RA		01	CB	3434
037800	C		LD	DX2	RB,X		01	CC	8934
037900	C		SRA	RB	RB		01	CD	3801
038000	C		ADD	RA	RB		01	CE	4100
038100	C		LL		FD RC * -3		01	CF	92FD
038200	C		ST	TEMP	RE		01	D0	A449
038300	C		LL		DO RE		01	D1	9400
038400	C		LK	SHFTR	RI		01	D2	F790
038500	C		ADD	X2	RB,X * X(2,2)=X(2,2)+		01	D3	4924
038600	C		ST	X2	RB,X * (DX(1,2)+DX(2,2))/16		01	D4	A924
038630	C		ST	PG	RE		01	D5	A447
038660	C		ST	PL	RL		01	D6	A548
038700	C		LL		OF RA * CLEAR FLAGS		01	D7	900F

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
038800	C		AND	RE	RA		01	D8	0004
038900	C		ST	RE	RA		01	D9	A004
	C		LL		00 RA		01	DA	9000
	C		LL		00 RC		01	DB	9200
	C		LD	DX3	RB		01	DC	8136
	C		AND	RB	RB		01	DD	0101
	C		BN/	HOP3			01	DE	D9E0
	C		LL		FF RA		01	DF	90FF
	C	HOP3	ST	TP	RB		01	E0	A15F
	C		LD	DX3	RB,X		01	E1	8936
	C		AND	RB	RB		01	E2	0101
	C		BN/	HOP4			01	E3	D9E5
	C		LL		FF RC		01	E4	92FF
	C	HOP4	ADD	TP	RB		01	E5	415F
	C		ADD	RC	RA,MP		01	E6	4402
039400	C		SLA	RB	RB		01	E7	3101
039500	C		SLL	RA	RA		01	E8	3800
039600	C		ADD	X3L	RB,X		01	E9	4926
039700	C		ADD	X3U	RA,X,MP		01	EA	4C28
039800	C		ST	X3U	RA,X * X(2,3)=X(2,3)+		01	EB	A828
039900	C		ST	X3L	RB,X * (DX(1,3)+DX(2,3))*2		01	EC	A926
039930	C		ST	PG	RE		01	ED	A447
039960	C		ST	PL	RL		01	EE	A548
040000	C		LD	DX4	RB,X		01	EF	8938

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
040100	C		SRA	RB	RB		01	F0	3501
040200	C		SRA	DX4	RA		01	F1	3438
040300	C		ADD	RA	RB		01	F2	4100
040400	C		LL		FD RC	* -3	01	F3	92FD
040500	C		ST	TEMP	RE		01	F4	4449
040600	C		LL		00 RE		01	F5	9400
040700	C		LK	SHFTR	RI		01	F6	F790
040800	C		ADD	X4	RB,X	* DX(2,4)=DX(2,4)+	01	F7	492A
040900	C		ST	X4	RB,X	* (DX(1,4)+DX(2,4))/16	01	F8	A92A
040930	C		ST	PG	RE		01	F9	A447
040960	C		ST	PL	RL		01	FA	A548
041000	C		SRA	DX5U	RA		01	FB	343C
041100	C		SRL	DX5L	RB		01	FC	3D3A
041200	C		LD	DX5U	RC,X		01	FD	8A3C
041300	C		SRA	RC	RC		01	FE	3602
041400	C		ST	TEMP	RC		01	FF	A249
041500	C		LD	DX5L	RC,X		02	00	8A3A
041600	C		SRL	RC	RC		02	01	3E02
041700	C		ADD	RC	RB		02	02	4102
041800	C		ADD	TEMP	RA,MP		02	03	4449
041900	C		LD	RA	RB		02	04	8100
042000	C		LL		FC RC	* -4	02	05	92FC
042100	C		ST	TEMP	RE		02	06	A449
042200	C		LL		00 RE		02	07	9400

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
042300	C		LK	SHFTR	RI		02	08	F790
042400	C		ADD	X5	RB,X	* X(2,5)=X(2,5)+	02	09	492C
042500	C		ST	X5	RB,X	* (DX(1,5)+DX(2,5))/8192	02	0A	A92C
	C		ST	PG	RE		02	0B	A447
042520	C		ST	PL	RL		02	0C	A548
042530	C		LD	DX6	RB,X		02	0D	893E
042540	C		SRA	RB	RB		02	0E	3501
042550	C		SRA	DX6	RA		02	0F	343E
042560	C		ADD	RA	RB		02	10	4100
042570	C		LL		FD RC	* -3	02	11	92FD
042580	C		ST	TEMP	RE		02	12	A449
042590	C		LL		00 RE		02	13	9400
042600	C		LK	SHFTR	RI		02	14	F790
042610	C		ADD	X6	RB,X	* X(2,6) = X(2,6)	02	15	492E
042620	C		ST	X6	RB,X	* (DX(1,6) DX(2,6))/16	02	16	A92E
042630	C		ST	PL	RL		02	17	A548
042640	C		LD	DX7	RB,X		02	18	8940
042700	C		SRA	RB	RB		02	19	3501
042800	C		SRA	DX7	RA		02	1A	3440
042900	C		ADD	RA	RB		02	1B	4100
043000	C		LL		FB RC	* -5	02	1C	92FB
043100	C		ST	TEMP	RE		02	1D	A449
043200	C		LL		00 RE		02	1E	9400
043300	C		LK	SHFTR	RI		02	1F	F790

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
043400	C		ADD	X7	RB,X	* X(2,7)=X(2,7)+	02	20	4930
043500	C		ST	X7	RB,X	* (DX(1,7)+DX(2,7))/64	02	21	A930
043600	C		J	INT			02	22	E0D8
	C	OUTPUT	LD	DX6	RB,X		02	23	893E
043800	C		LL		FD RC	* -3	02	24	92FD
043900	C		ST	TEMP	RE		02	25	A449
044000	C		LL		00 RE		02	26	9400
044100	C		LK	SHFTR	RI		02	27	F790
044200	C		ADD	V2	RB		02	28	411D
044300	C		ST	Y	RB	* Y=V2+DX(2,6)/8	02	29	A112
	C		LL		33 RC		02	2A	9233
	C		ST	TEMP	RE		02	2B	A449
	C		LL		00 RE		02	2C	9400
	C		LK	MULT	RI		02	2D	F75B
	C		LL		FA RC		02	2E	92FA
	C		ST	TEMP	RE		02	2F	A449
	C		LL		00 RE		02	30	9400
	C		LK	DPSH	RI		02	31	F778
	C		ST	YMATH	RB		02	32	A110
	C		ST	YTEMP	RB		02	33	A161
044400	C		J	DIAG			02	34	E2A5
044600	C	MANG	SLA	SWSET	RA		02	35	3009
	C		BC/	CHKD		* BR IF SW A SET	02	36	D87D
044800	C		LL		10 RA		02	37	9010

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
044900	C		AND	SWSET	RA		02	38	0009
045100	C		BZ	CHKC		* BR IF SW D SET	02	39	CABC
			LL		04 RA		02	3A	9004
045200	C		OR	DEABLE	RA		02	3B	1043
045300	C		ST	DEABLE	RA	* DISABLE SERVO=A	02	3C	A043
045400	C		LL		02 RA		02	3D	9002
045500	C		AND	VALID	RA		02	3E	0009
045700	C		BZ	CUTB		* BR IF SERVO=B NOT VALID	02	3F	CA49
			LL		80 RA		02	40	9080
045800	C		OR	SETSW	RA		02	41	1043
045900	C		ST	SETSW	RA	* SHUTTLE SW=A	02	42	A043
046000	C		ST	PAGE	RE		02	43	AA44
046100	C		LK	EXIT5	RC	* LOCATION COUNTER ST IN RC	02	44	F290
046200	C		LL		80 RA		02	45	9080
046300	C		AND	SWSET	RA		02	46	0009
046600	C		BZ/	SERV		* BR IF SW=A NOT SET	02	47	DA4C
			J	EX5			02	48	E28D
046700	C		OR	DEABLE	RA		02	49	9002
046800	C		ST	DEABLE	RA	* DISABLE SERVO=B	02	4A	1043
046900	C		LL		01 RA		02	4B	A043
			C	SERV			02	4C	9001
047000	C		AND	VALID	RA		02	4D	0009
047200	C		BZ	KILL		* BR IF SERVO=C NOT VALID	02	4E	CA57
			LL		10 RA		02	4F	9010

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR ACI	P	D	CODE
047300	C		OR	SETSW	RA		02	50	1043
047400	C		ST	SETSW	RA	* SHUTTLE SW=D	02	51	A043
047500	C		ST	PAGE	RE		02	52	A444
047600	C		LK	EXITS	RC		02	53	F290
047700	C		LL		10 RA		02	54	9010
047800	C		AND	SWSET	RA		02	55	0009
	C		BZ	EX5		* BR IF SW=D SET	02	56	CASD
048000	C	KILL	LL		07 RA		02	57	9007
048100	C		OR	DEABLE	RA		02	58	1043
048200	C		ST	DEABLE	RA	* DISABLE ALL SERVOS	02	59	A043
	C		ST	STATUS	RA		02	5A	A00A
048300	C		W				02	5B	E25B
048400	C	CHKC	LL		20 RA		02	5C	9020
048500	C		AND	SWSET	RA		02	5D	0009
	C		BZ	CHKSA		* BR IF SW=C SET	02	5E	CA62
048700	C		LL		21 RA		02	5F	9021
048800	C		OR	SETSW	RA		02	60	1043
048900	C		ST	SETSW	RA	* DISABLE SERVO=C AND SET SW=C	02	61	A043
049000	C	CHKSA	LL		04 RA		02	62	9004
049100	C		AND	VALID	RA		02	63	0009
	C		BZ	CUTA		* BR IF SERVO=A NOT VALID	02	64	CA6E
049300	C	SHUTB	LL		40 RA		02	65	9040
049400	C		OR	SETSW	RA		02	66	1043
049500	C		ST	SETSW	RA	* SHUTTLE SW=B	02	67	A043

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
049600	C		ST	PAGE	RE		02	68	A444
049700	C		LK	EXIT5	RC		02	69	F290
049800	C		LL		40 RA		02	6A	9040
049900	C		AND	SWSET	RA		02	6B	0009
	C		BZ	EX5		* BR IF SW-B SET	02	6C	CA8D
050100	C		J	KILL			02	6D	E257
050200	C	CUTA	LL		04 RA		02	6E	9004
050300	C		OR	DEABLE	RA		02	6F	1043
050400	C		ST	DEABLE	RA	* DISABLE SERVO-A	02	70	A043
050500	C		LL		02 RA		02	71	9002
050600	C		AND	VALID	RA		02	72	0009
	C		BZ	KILL		* BR IF SERVO-B NOT VALID	02	73	CA57
050800	C		LL		80 RA		02	74	9080
050900	C		OR	SETSW	RA		02	75	1043
051000	C		ST	SETSW	RA	* SHUTTLE SW-A	02	76	A043
051100	C		ST	PAGE	RE		02	77	A444
051200	C		LK	EXIT5	RC		02	78	F290
051300	C		LL		80 RA		02	79	9080
051400	C		AND	SWSET	RA		02	7A	0009
	C		BZ	SHUTB		* BR IF SW-A SET	02	7B	CA65
051600	C		J	KILL			02	7C	E257
051700	C	CHKD	LL		10 RA		02	7D	9010
051800	C		AND	SWSET	RA		02	7E	0009
	C		BZ/	CUTB		* BR IF SW-D NOT SET	02	7F	DA49

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR ACI	P	D	CODE
052000	C		LL		20 RA		02	80	9020
052100	C		AND	SWSET	RA		02	81	0009
C	BZ	SERV				* BR IF SW-C SET	02	82	CAB6
052300	C		LL		21 RA		02	83	9021
052400	C		OR	SETSW	RA		02	84	1043
052500	C		ST	SETSW	RA	* DISABLE SERVO-C AND SET SW-C	02	85	A043
052600	C	SERV	LL		02 RA		02	86	9002
052700	C		AND	VALID	RA		02	87	0009
C	BZ	KILL				* BR IF SERVO-B NOT VALID	02	88	CA57
052900	C		LL		'80 RA		02	89	9080
053000	C		OR	SETSW	RA		02	8A	1043
053100	C		ST	SETSW	RA	* SHUTTLE SW-A	02	8B	A043
053200	C		J	SHJTB			02	8C	E265
C	EX5	LL			35- RC		02	8D	9202-9235
C		ST	PLACE	RC			02	8E	A25D
C		J	MANG				02	8F	E235
053300	C	EXIT5	ST	PLACE	RC		02	90	A25D
C		LL			10 RA		02	91	9010
C		ST	SAMPLE	RA			02	92	A042
C		LL			FB RA		02	93	90FB
C		ST	FAILS	RA			02	94	A060
053400	C		J	PAUSE			02	95	E2A1
053430	C	COMP	ST	PG	RE		02	96	A447
053460	C		ST	PL	RL		02	97	A548

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
	C		LD	YTEMP	RA		02	98	8061
	C		ADD	YHAUL	RA		02	99	400E
053700	C		BP	SKIP			02	9A	C89C
053800	C		TC	RA	RA		02	9B	6000
053900	C	SKIP	SUB	TOL	RA		02	9C	705A
	C		BN	PAUSE			02	9D	C9A1
	C		INC	FAILS	RA		02	9E	5060
	C		ST	FAILS	RA		02	9F	A060
	C		BZ	PLACE	I		02	A0	C25D
	C	PAUSE	LL		08 RA		02	A1	9008
	C		OR	SETSW	RA		02	A2	1043
	C		ST	STATUS	RA		02	A3	A00A
054300	C	WATE	W			* WAIT FOR NEXT INPUT COMMAN	02	A4	E2A4
056430	C	DIAG	ST	PG	RE		02	A5	A447
056460	C		ST	PL	RL		02	A6	A548
056500	C		LL		FD RA	* -3	02	A7	90FD
056600	C		ST	TEST	RA		02	A8	A045
058600	C	INCTCM	LL		00 RA	* INC TC MP	02	A9	9000
058700	C		LL		00 RB		02	AA	9100
058800	C	STAR2	INC	RA	RA		02	AB	5000
058900	C		INC	RB	RB,MP		02	AC	5501
059000	C		TC	RA	RC		02	AD	6200
	C		XC/	STAR2			02	AE	8B48
059100	C		TC	RB	RX,MP		02	AF	6701

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
	C		TC	RB	RB		02	B0	6101
059400	C		CPA	RA	RC		02	B1	2200
059500	C		CPA	RB	RX,MP		02	B2	2701
059600	C		BZ/	ERR2			02	B3	DAFO
059700	C	CNTR0U	LL		F9 RA * CNT , -7		02	B4	90F9
059800	C		LL		F8 RB * -8		02	B5	91F8
059900	C		INC	RB	RB		02	B6	5101
060100	C	STAR4	BZ	ERR2			02	B7	CAFO
060200	C		CNT	STAR4	RA		02	B8	80B7
	C		INC	RB	RB		02	B9	5101
	C		BZ/	ERR2			02	BA	DAFO
	C	AD	LL		00 RA * ADD SUB SLL SLA SRL SRA		02	BB	9000
	C		LL		9D RB		02	BC	919D
	C		LL		13 RC		02	BD	9213
	C		ADD	RC	RB		02	BE	4102
	C		LL		DE RC		02	BF	9207
	C		SUB	RC	RB		02	C0	7102
	C		LL		A2 RC		02	C1	92A2
	C		ADD	RC	RB		02	C2	4102
	C		BO/	ERR3		✓4B	02	C3	DCP4
	C		SLA	RB	RB		02	C4	3101
	C		SRA	RB	RB		02	C5	3501
	C		SLL	RB	RB		02	C6	3901
	C		SLL	RA	RA		02	C7	3800

P/L	CD REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
C		SLA	RB	RB		02	CB	3101
C		SLL	RA	RA		02	C9	3800
C		SRA	RB	RB		02	CA	3501
C		SRL	RB	RB		02	CB	3001
C		CPA	RA	RB		02	CC	2100
C		BZ/	ERR3			02	CD	DAFA
068300	C	LOGIC	LL	55 RA	* AND OR EOR	02	CE	9055
068400	C	LL		33 RB		02	CF	9133
C		AND	RB	RA		02	DO	0001
068600	C	LL		55 RC		02	D1	9255
068700	C	EOR	RC	RB		02	D2	1502
068800	C	ADD	RB	RA		02	D3	4001
068900	C	LL		33 RB		02	D4	9133
069000	C	OR	RC	RB		02	D5	1102
069100	C	SUB	RB	RA		02	D6	7001
C		BZ/	ERR11			02	D7	DAF8
069300	C	LINK	LL	00 RA		02	D8	9000
069400	C	LL		00 RB		02	D9	9100
069500	C	LL		00 RC		02	DA	9200
069600	C	LL		02 RX		02	DB	9302
069700	C	LKCK	INC	RC		02	DC	5202
069800	C	XC/	XTEST			02	DD	38E2
069900	C	ADD	RX	RB	* ADD 2	02	DE	4103
070000	C	CPA	RB	RC	* COMPARE RB RC SHOULD BE 0	02	DF	2201

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
070100	C		BZ	RI	I	* INDIRECT BR IF COMPARE	02	E0	C207
070200	C		J	ERR11			02	E1	E2F8
070300	C	XTEST	INC	RA	RA,MP		02	E2	5400
	C		LK	LKCK	RI		02	E3	F7DC
070500	C	LKCHK	INC	RA	RA		02	E4	5000
070600	C	CPA	RX	RA	RA	* COMPARE RA TO 2	02	E5	2003
070700	C	XZ/	ERR11				02	E6	8AF8
	C	LD	SAMPLE	RA	RA		02	E7	8042
	C	AND	RA	RA	RA		02	E8	0000
	C	BZ	JCOMP				02	E9	CAED
	C	INC	RA	RA	RA		02	EA	5000
	C	ST	SAMPLE	RA	RA		02	EB	A042
073400	C	J	PAUSE			* FIRST 127 SAMPLES NOT COMPLETE	02	EC	E2A1
	C	JCOMP	LL		00 RA		02	ED	9000
	C	ST	SAMPLE	RA	RA		02	EE	A042
073450	C	J	COMP				02	EF	E296
073900	C	ERR2	INC	TEST	RA		02	FO	5045
074000	C	ST	TEST	TEST	RA		02	F1	A045
074100	C	BZ/	INCTCM				02	F2	DAA9
074200	C	W					02	F3	E2F3
074300	C	ERR3	INC	TEST	RA		02	F4	5045
074400	C	ST	TEST	TEST	RA		02	F5	A045
	C	BZ/	AD				02	F6	DAB8
074600	C	W					02	F7	E2F7

P/L	CD	REF	OP	OPERAND	MODIFIERS	COMMENTS OR BCI	P	D	CODE
077500	C	ERR11	INC	TEST	RA		02	F8	5045
077600	C		ST	TEST	RA		02	F9	A048
077300	C		BZ/	LOGIC			02	FA	DACE
077800	C		W				02	FB	E2F8

002 ERROR MSG

* * *

DATA TAGS

BEGAN	01 1C	01 1D	01 1E	01 1F	01 1G	01 1H	01 1I	01 1J	01 1K	01 1L	01 1M	01 1N	01 1O	01 1P	01 1Q	01 1R	01 1S	01 1T	01 1U	01 1V	01 1W	01 1X	01 1Y	01 1Z	01 1AA	01 1AB	01 1AC	01 1AD	01 1AE	01 1AF	01 1AG	01 1AH	01 1AI	01 1AJ	01 1AK	01 1AL	01 1AM	01 1AN	01 1AO	01 1AP	01 1AQ	01 1AR	01 1AS	01 1AT	01 1AU	01 1AV	01 1AW	01 1AX	01 1AY	01 1AZ	01 1BA	01 1BB	01 1BC	01 1BD	01 1BE	01 1BF	01 1BG	01 1BH	01 1BI	01 1BJ	01 1BK	01 1BL	01 1BM	01 1BN	01 1BO	01 1BP	01 1BQ	01 1BR	01 1BS	01 1BT	01 1BU	01 1BV	01 1BW	01 1BX	01 1BY	01 1BZ	01 1CA	01 1CB	01 1CC	01 1CD	01 1CE	01 1CF	01 1CG	01 1CH	01 1CI	01 1CJ	01 1CK	01 1CL	01 1CM	01 1CN	01 1CO	01 1CP	01 1CQ	01 1CR	01 1CS	01 1CT	01 1CU	01 1CV	01 1CW	01 1CX	01 1CY	01 1CZ	01 1DA	01 1DB	01 1DC	01 1DD	01 1DE	01 1DF	01 1DG	01 1DH	01 1DI	01 1DJ	01 1DK	01 1DL	01 1DM	01 1DN	01 1DO	01 1DP	01 1DQ	01 1DR	01 1DS	01 1DT	01 1DU	01 1DV	01 1DW	01 1DX	01 1DY	01 1DZ	01 1EA	01 1EB	01 1EC	01 1ED	01 1EE	01 1EF	01 1EG	01 1EH	01 1EI	01 1EJ	01 1EK	01 1EL	01 1EM	01 1EN	01 1EO	01 1EP	01 1EQ	01 1ER	01 1ES	01 1ET	01 1EU	01 1EV	01 1EW	01 1EX	01 1EY	01 1EZ	01 1FA	01 1FB	01 1FC	01 1FD	01 1FE	01 1FF	01 1FG	01 1FH	01 1FI	01 1FJ	01 1FK	01 1FL	01 1FM	01 1FN	01 1FO	01 1FP	01 1FQ	01 1FR	01 1FS	01 1FT	01 1FU	01 1FV	01 1FW	01 1FX	01 1FY	01 1FZ	01 1GA	01 1GB	01 1GC	01 1GD	01 1GE	01 1GF	01 1GG	01 1GH	01 1GI	01 1GJ	01 1GK	01 1GL	01 1GM	01 1GN	01 1GO	01 1GP	01 1GQ	01 1GR	01 1GS	01 1GT	01 1GU	01 1GV	01 1GW	01 1GX	01 1GY	01 1GZ	01 1HA	01 1HB	01 1HC	01 1HD	01 1HE	01 1HF	01 1HG	01 1HH	01 1HI	01 1HJ	01 1HK	01 1HL	01 1HM	01 1HN	01 1HO	01 1HP	01 1HQ	01 1HR	01 1HS	01 1HT	01 1HU	01 1HV	01 1HW	01 1HX	01 1HY	01 1HZ	01 1IA	01 1IB	01 1IC	01 1ID	01 1IE	01 1IF	01 1IG	01 1IH	01 1II	01 1IJ	01 1IK	01 1IL	01 1IM	01 1IN	01 1IO	01 1IP	01 1IQ	01 1IR	01 1IS	01 1IT	01 1IU	01 1IV	01 1IW	01 1IX	01 1IY	01 1IZ	01 1JA	01 1JB	01 1JC	01 1JD	01 1JE	01 1JF	01 1JG	01 1JH	01 1JI	01 1JJ	01 1JK	01 1JL	01 1JM	01 1JN	01 1JO	01 1JP	01 1JQ	01 1JR	01 1JS	01 1JT	01 1JU	01 1JV	01 1JW	01 1JX	01 1JY	01 1JZ	01 1KA	01 1KB	01 1KC	01 1KD	01 1KE	01 1KF	01 1KG	01 1KH	01 1KI	01 1KJ	01 1KK	01 1KL	01 1KM	01 1KN	01 1KO	01 1KP	01 1KQ	01 1KR	01 1KS	01 1KT	01 1KU	01 1KV	01 1KW	01 1KX	01 1KY	01 1KZ	01 1LA	01 1LB	01 1LC	01 1LD	01 1LE	01 1LF	01 1LG	01 1LH	01 1LI	01 1LJ	01 1LK	01 1LL	01 1LM	01 1LN	01 1LO	01 1LP	01 1LQ	01 1LR	01 1LS	01 1LT	01 1LU	01 1LV	01 1LW	01 1LX	01 1LY	01 1LZ	01 1MA	01 1MB	01 1MC	01 1MD	01 1ME	01 1MF	01 1MG	01 1MH	01 1MI	01 1MJ	01 1MK	01 1ML	01 1MN	01 1MO	01 1MP	01 1MQ	01 1MR	01 1MS	01 1MT	01 1MU	01 1MV	01 1MW	01 1MX	01 1MY	01 1MZ	01 1NA	01 1NB	01 1NC	01 1ND	01 1NE	01 1NF	01 1NG	01 1NH	01 1NI	01 1NJ	01 1NK	01 1NL	01 1NM	01 1NO	01 1NP	01 1NQ	01 1NR	01 1NS	01 1NT	01 1NU	01 1NV	01 1NW	01 1NX	01 1NY	01 1NZ	01 1OA	01 1OB	01 1OC	01 1OD	01 1OE	01 1OF	01 1OG	01 1OH	01 1OI	01 1OJ	01 1OK	01 1OL	01 1OM	01 1ON	01 1OO	01 1OP	01 1OQ	01 1
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	------

YTEMP 01 21

CMD TAGS

AD	02 88	BEGIN	00 96
CCC	00 86	CHKC	02 5C
CHKD	02 7D	CHKSA	02 62
CLR	00 24	CNTRDU	02 B4
COMP	02 96	CONT	00 1F
CONT1	00 1D	CONTIN	01 59
CUTA	02 6E	CUTB	02 49
DIAG	02 A5	DPSH	00 78
EAR	00 16	FIN	00 1A
ERR	00 8A	ERR11	02 F8
ERR2	02 F0	ERR3	02 F4
EX5	02 8D	EXIT2	00 94
EXIT3	00 88	EXIT5	02 90
HOP1	01 BA	HOP2	01 BF
HOP3	01 EQ	HOP4	01 E8
HOP5	00 9F	HOP6	00 B4
HOP7	01 38	HOP8	00 0A
INCTCM	02 A9	INT	00 D5
JCOMP	02 ED	KILL	02 57
LINK	02 D8	LCHK	02 E4
LCKK	02 DC	LOGIC	02 CE
LOOP1	00 63	LOOP2	00 66
LOOP3	00 6E	LOOP4	00 74
LP5	01 39	MANG	02 35
MULT	00 58	NEG1	00 8E
OUTPUT	02 23	PAGE1	01 00
PAR	00 10	PAUSE	02 A1
PLUS	00 84	PUT	01 AE
QCAL	01 21	QNEG	01 52
REPEAT	00 EA	RETURN	00 68
RUPT	00 02	SERV8	02 86
SERV6	02 4C	SHFTR	00 90
SHUTB	02 65	SKIP	02 9C
SKP	00 0D	STAR2	02 AB
STAR4	02 B7	START	00 28
VARI	01 AF	WATE	02 A4
XTEST	02 E2		

APPENDIX D

SCALING EXAMPLE

D-1

APPENDIX D

Scaling Example

The following is the procedure used in determining the scaling requirements for the equation

$$(1) \quad X(1,1) = X(2,1) + T * DX(2,1)$$

T is the time between samples or $\frac{1}{256} = 2^{-8}$ seconds

First the implicit scale for each of the variables must be determined.

Since the computer only does integer arithmetic, the absolute value of each variable must be scaled in such a manner as to obtain maximum accuracy in the 8-bit processor.

Maximum values were obtained for each variable at an 8 degree input using floating point arithmetic on the IBM 1130 computer. These are listed below :

$$X(K,1) \leq 3.82$$

$$DX(K,1) \leq 19.25$$

$X(K,1)$ is a double precision variable (16 bits) .

$$3.82_{10} = 3.643656_8 = 011.110100011110101110_2$$

For 16 bit accuracy, the binary word can be written as

$$0111101000111101_2 \times 2^{-13}$$

-13 will then be the implicit binary exponent for $X(K,1)$.

Likewise, for $DX(K,1)$ (single precision 8 bits)

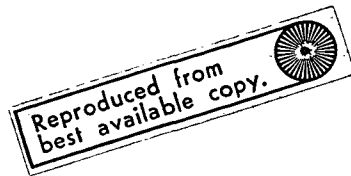
$$19.25_{10} = 23.2_8 = 010011.010_2 = 01001101_2 \times 2^{-2}$$

-2 will then be the implicit binary exponent for $DX(K,1)$.

From equation (1) we write:

$$\begin{aligned}X(1,1)*2^{-13} &= X(2,1)*2^{-13} + 2^{-8} *DX(2,1) *2^{-2} \\&= \left[X(2,1) + 8 *DX(2,1) \right] 2^{-13} \\X(1,1) &= X(2,1) + 8 *DX(2,1)\end{aligned}$$

APPENDIX E



PAPER TAPE GENERATION

E-1

CONFIDENTIAL


```

// DUP
*STORE      WS  UA  DW057
// FOR
*LIST ALL
*ONE WORD INTEGERS
*IOCS(CARD)
  INTEGER A(16)
  DIMENSION INA(161)
  DATA A /'0','1','2','3','4','5','6','7','8','9','A','B','C','D',
1'E','F'/
  DATA M1,M2,NN /'J','K','L'/
  KNT = 10
  M = 1
3 CONTINUE
  KCNT = 0
15 IF(KCNT-767) 5,5,10
5 NPAG = KCNT / 256
  NDIS = KCNT-NPAG*256
  NPAG = NPAG+1
  INA(3)=A(NPAG)
  N1 = NDIS / 16
  N2 = NDIS-N1*16+1
  N1 = N1 + 1
  INA(2)=A(1)
  INA(1)=M1
  INA(10)=M2
  INA(4) = A(N1)
  INA(5) = A(N2)
  GO TO (50,55),M
50 CONTINUE
  DO 4 I=6,9
4 INA(I) = A(11)
  GO TO 60
55 CONTINUE
  DO 30 I=6,9
30 INA(I) = A(6)
60 CONTINUE
  CALL DW054 (INA,KNT)
  KCNT = KCNT + 1
  GO TO 15
10 GO TO (20,25),M
20 M = M + 1
  GO TO 3
25 DO 35 I=2,9
35 INA(I) = NN
  INA(1)=M1
  INA(10)=M2
  CALL DW054 (INA,KNT)
  CALL EXIT
  END
//XEQ

```

// ASM

*LIST

*XREF

* SUBROUTINES CALLABLE BY FORTRAN PROGRAMS TO READ	DW510020
	DW510035
	DW510040
* PAPER TAPE IN ASCII-8 CODE AND CONVERT TO EBCDIC	DW510050
* CODE AND CONVERT EBCDIC CODE TO ASCII-8 CODE AND	DW510060
* PUNCH PAPER TAPE. THE PUNCH ROUTINE HAS ITS OWN	DW510070
* INTERNAL BUFFER SO THAT PUNCHING IS OVERLAPPED	DW510080
* WITH PROCESSING. THE READ ROUTINE IS STRUCTURED	DW510090
* SO THAT THE USER CAN OVERLAP THE INPUT IF DE-	DW510100
* SURED, HOWEVER, HE MUST ALLOCATE THE BUFFER AREAS	DW510110
* TO PROVIDE THE OVERLAP.	DW510120
* THE PROGRAM AREAS ENCLOSED BY ASTERISKS (**...**)	DW510130
* SHOULD BE ANALYZED CAREFULLY AND MODIFIED APP-	DW510140
* PROPRIATELY IF IT IS DESIRED TO CHANGE THE BUFFER	DW510150
* SIZE.	DW510160
	DW510170
ENT DW051 CHECK TO DETERMINE WHETHER	DW510180
1134 PAPER TAPE READER IS	DW510190
BUSY.	DW510200
	DW510210
	DW510220
* DW051 CALLING SEQUENCE IS AS FOLLOWS	DW510230
	DW510240
	DW510250
CALL DW051	DW510260
(DW051 WAITS IN A PROGRAMMED LOOP UN-	DW510270
TIL THE PAPER TAPE READER IS NOT	DW510280
RJSY.)	DW510290
ENT DW052 INITIATE READING OF PAPER	DW510300
TAPE CHARACTERS.	DW510310
	DW510320
* DW052 CALLING SEQUENCE IS AS FOLLOWS	DW510330
	DW510340
CALL DW052 (IN, I, IS)	DW510350
WHERE,	DW510360
IN = THE INPUT ARRAY (MUST BE	DW510370
DIMENSIONED TO A SIZE	DW510380
OF 161).	DW510390
	DW510400
I = THE MAXIMUM NUMBER OF	DW510410
CHARACTERS TO BE CONVERTED	DW510420
	DW510430
IS = A PROGRAM SWITCH WHICH IS	DW510440
SFT TO 2 IF THE ISS DETECT	DW510450
ED A POST-OPERATIVE ERROR.	DW510460
	DW510470
ENT DW053 CONVERT PAPER TAPE CHAR-	DW510480
ACTERS TO EBCDIC	DW510490
	DW510490

```

* DW053 CALLING SEQUENCE IS AS FOLLOWS
*
* CALL DW053 ( IN, KOUNT )
*
* WHERE,
* IN = THE ARRAY FILLED WITH
* PAPER TAPE CHARACTERS
* BY DW052
*
* KOUNT = THE NUMBER OF EBCDIC
* CHARACTERS IN THE 'IN'
* ARRAY.
*
* ENT DW054 CONVERT CHARACTERS TO
* ASCII-8 AND PUNCH TAPE.
*
* DW054 CALLING SEQUENCE IS AS FOLLOWS
*
* CALL DW054 ( IOUT, KOUNT )
* WHERE,
* IOUT = AN ARRAY CONTAINING
* EBCDIC CHARACTERS TO BE
* PUNCHED
*
* KOUNT = COUNT OF NUMBER OF OUT-
* PUT CHARACTERS.
*
* DW055 IS A FORTRAN CALLABLE SURROUTINE WHICH
* ALLOWS DYNAMIC MODIFICATION OF THE 'STOP' AND
* 'DELETE' CHARACTERCODES OF THE IBM INTERRUPT
* SERVICE SURROUTINE.
*
* ENT DW055 MODIFY 'DFL' AND 'STOP'
* CHARACTERS IN 'PAPTX' AND
* DW053.
*
* DW055 CALLING SEQUENCE IS AS FOLLOWS
*
* CALL DW055 ( IDEL, ISTOP, IERC )
*
* WHERE,
* IDEL = THE DESIRED DELETE CHAR-
* ACTER IN HEX (BITS 0-7)
*
* ISTOP = THE DESIRED PAPER TAPE
* CODE STOP CHARACTER IN
* HEX (BITS 0-7)
*
* IERC = THE DESIRED STOP CODE

```

CHARACTER AFTER IT HAS

DW510961

BEEN TRANSFORMED BY

DW510962

THE CODE CONVERSION ROUTINE, IN EBCDIC (BITS

DW510963

0-7 OR 'A1' FORMAT).

DW510964

DW510965

* DW510970

* DW510980

DW510990

DW511000

DW511010

DW511020

DW511030

DW511040

DW511050

DW511060

DW511070

DW511080

DW511090

DW511100

DW511110

DW511120

DW511130

DW511140

DW511150

DW511160

DW511170

DW511180

DW511190

DW511200

DW511210

DW511220

DW511230

DW511240

DW511250

DW511260

DW511270

DW511280

DW511290

DW511300

DW511310

DW511320

DW511330

DW511340

DW511350

DW511360

DW511370

DW511380

DW511390

DW511400

DW511410

DW511420

DW511430

DW511440

DW511450


```
USAVE DC *--* LOCAL ENTRY POINT DW511880
SAVE1 LDX L1 *--* RESTORE INDEX REGISTER 1. DW511890
SAVE2 LDX L2 *--* RESTORE INDEX REGISTER 2. DW511900
LDD SAVEA RESTORE ACC AND EXT DW511910
LDS RESTORE TOGGLES DW511920
BSC I USAVE RETURN TO LOCAL PROGRAM. DW511930
* DW511940
DW053 DC *--* DW053 ENTRY POINT. DW511950
* BSI SAVE SAVE ACC, XR'S, TOGGLES. DW511960
LD I DW053 LOAD LOW-ORDER ARRAY DW511970
STO L 2 ADDRESS IN INDEX REG. 2. DW511980
STX L2 BASE SAVE BASE ADDRESS FOR DW511990
* CHARACTER COUNT CALC. DW512000
* STX 2 ADD3+1 SET UP LOW ORDER OUTPUT DW512010
* ADDRESS. DW512020
***** DW512030
MDX 2 -80 CALCULATE AND STORE HIGH- DW512040
STX 2 ADD6+1 ORDER INPUT ADDRESS. DW512050
***** DW512060
MDX 2 -78 CALCULATE DW512070
***** DW512080
STX 2 ADD5+1 AND STORE DW512090
MDX 2 -1 HIGH ORDER OUTPUT ADDRESS. DW512100
STX 2 ADD1 STORE INPUT AND OUTPUT DW512110
STX 2 ADD2 ADDRESSES FOR 'ZIPCO', AND DW512120
* STX 2 ADD4+1 LOW ORDER INPUT ADDRESS DW512130
* FOR PROGRAM. DW512140
LIBF ZIPCO CONVERT FROM ASCII-8 ITY DW512150
DC 70000 CODE TO EBCDIC. BOTH INPUT DW512160
ADD1 DC *--* AND OUTPUT ARE PACKED. DW512170
ADD2 DC *--* SPECIAL CODE TABLE PRO- DW512180
***** DW512190
DC 160 VIED BY DW056. I/O ADD- DW512200
***** DW512210
CALL DW056 RESSES CALCULATED BY PROG. DW512220
LD L NOP SET PROGRAM SWITCH TO RE- DW512230
STO END TURN CHARACTER COUNT TO DW512240
* CALLING PROGRAM. DW512250
MDX 2 -1 CALCULATE ADDRESS OF CHAR- DW512260
LD 2 ACTER COUNT SET BY DW052 DW512270
* AND PLACE COUNT IN ACC. DW512280
* STO CHCNT SAVE CHARACTER COUNT. DW512290
***** DW512300
SKP E
S COO1
STO CNTR1
MDX 2 +1
LOOP1 SLT 32
LD 2
RTE 8
```


ADD5	LDX	L1	**	END	UNPACKING DONE, WRAP-UP.	DW512580
ADD6	LD	L	**		SET UP HIGH ARRAY ADDRESS.	DW512590
					INPUT TO ACCUMULATOR.	DW512600
	LDX	2	1		SET EVEN/ODD	DW512610
	STX	2	ODD		SWITCH TO 'ODD'.	DW512620
	BSI		UNPAK		UNPACK CHARACTERS AND STO.	DW512630
	MDX	1	+4		CALCULATE AND SAVE NEW	DW512640
	STX	1	ADD5+1		HIGH ORDER OUTPUT ADDRESS.	DW512650
	MDM		ADD6+1,-1		CALCULATE NEW INPUT ADD.	DW512660
					* COUNT DECREMENT INSTRUCTION MAY BE PLACED HERE.	DW512670
					* DEPENDING ON THE BUFFER SIZE.	DW512680
END	B	DC	**	ADD3	CONTINUE UNPACK.	DW512690
			**		** PROGRAM SWITCH **	DW512700
	LD		STOPS		IS 'STOP' SWITCH	DW512710
	BNZ		NSTOP		TURNOED ON.	DW512720
	LD		BASE		YES, CALCULATE	DW512730
	S	L	1		CHARACTER COUNT.	DW512740
STORE	MDM		DW053,+1		CALCULATE CHARACTER COUNT	DW512750
	LDX	12	DW053		ADDRESS AND RETURN ARGU-	DW512760
	STO	12	0		MENT TO CALLING PROGRAM.	DW512770
	MDM		END,EXIT-END-1		SET PROGRAM SWITCH TO	DW512780
*					BY-PASS CHARACTER COUNT	DW512790
*					STORE ROUTINE.	DW512800
	LD		ODD		IF HIGH-ORDER WORDS ARE	DW512810
	BOD		UNPKA		BEING UNPACKED, CONTINUE	DW512820
*					OPERATION.	DW512830
EXIT	BSI	L	USAVE		RESTORE ACC, XR'S, TOGGLES	DW512840
	MDM		DW053,+1		RETURN TO	DW512850
	BSC	I	DW053		CALLING PROGRAM.	DW512860
NSTOP	LD		CHCNT		LOAD CHARACTER COUNT	DW512870
	B		STORE		SET BY 'DW052'.	DW512880
UNPAK	DC		**		LOCAL ROUTINE ENTRY POINT.	DW512890
	LDX	2	+2		SET UP ITERA-	DW512900
	STX	2	UNCNT		TION COUNT.	DW512910
	RTE		8		SECOND CHARACTER TO EXT.	DW512920
	SLA		8		LEFT-JUSTIFY CHARACTER.	DW512930
UNPK1	OR		BLANK		SET RIGHT HALF TO EBCDIC	DW512940
*					BLANK	DW512950
	STO	1	0		STORE CHARACTER.	DW512960
	MDX	1	-1		CALCULATE NEW STORE ADD.	DW512970
	EOR		STOP		IS CHARACTER	DW512980
	BNZ		UNPKA		A 'STOP'.	DW512990
	STO		STOPS		YES,TURN ON 'STOP'	DW513000
	B		END		SWITCH AND WRAP UP PROGRAM	DW513010
UNPKA	MDM		UNCNT,-1		DECREMENT ITERATION COUNT.	DW513020
	R		UNPK2			DW513030
	BSC	I	UNPAK		OPERATION COMPLETE, RETURN	DW513040
*					TO LOCAL CALLING PROGRAM.	DW513050
UNPK2	SLT		16		MOVE NEXT CHARACTER TO	DW513060
*					ACCUMULATOR	DW513070

B		UNPK1	UNPACK NEXT CHARACTER.	DW513080
* * * * *	* * * * *	DATA AND CONSTANTS AREA	* * * * *	DW513090
* * * * *	* * * * *			DW513100
* * * * *	* * * * *			DW513110
BASE DC	**	BASE ADDRESS.		DW513120
BLANK DC	/0040	EBCDIC BLANK CONSTANT		DW513130
CHCNT DC	**	CHARACTER COUNT SAVE AREA.		DW513140
CNT1 DC	**	ITERATION COUNTER.		DW513150
COUNT DC	**	CONVERT COUNTER.		DW513160
NOP MDX	*	INITIAL PROGRAM SWITCH.		DW513170
ODD DC	**	EVEN/ODD SWITCH.		DW513180
STOP DC	**	STOP CHARACTER (SET BY		DW513190
*		DW055).		DW513200
STOPS DC	**	STOP CHARACTER SWITCH.		DW513210
UNCNT DC	/0000	ITERATION COUNTER.		DW513220
PCHER DC	**	TAPE PUNCH ERROR ROUTINE.		DW513230
BSC 1	PCHER	RETRY PUNCH OPERATION.		DW513240
C002 DC	2	DECIMAL CONSTANT.		DW513250
RDRER DC	**	TAPE READER ERROR ROUTINE.		DW513260
LD	C002	ERROR SWITCH		DW513270
ERR STO L	**	TO DECIMAL 2.		DW513280
SRA	2	CLEAR ACCUMULATOR SO THAT		DW513290
BSC 1	RDRER	READ OPERATION IS TERM-		DW513300
*		INATED IMMEDIATELY.		DW513310
*				DW513320
DW054 DC	**	DW054 ENTRY POINT.		DW513330
*				DW513340
RSI L	SAVE	SAVE ACC. XR'S, TOGGLES.		DW513350
MDM	DW054,+1	CALCULATE ADD-		DW513360
LDX 11	DW054	RESS, CHECK		DW513370
BSI L	LIMIT	RANGE AND STORE CHAR COUNT		DW513380
STO	ADD9	FOR 'ZIPCO',		DW513390
STO	RUF	OUTPUT BUFFER, AND		DW513400
STO L	1	IN INDEX REGISTER 1.		DW513410
MDM	DW054,-1	CALCULATE AND		DW513420
LDX 12	DW054	LOAD LOW ORDER		DW513430
LD	2	ADDRESS IN ACC.		DW513440
S	ADD9	CALCULATE AND		DW513450
STO	ADD10+1	STORE LOW-ORDER ADD-		DW513460
STO	ADD11+1	RESS - CHARACTER COUNT.		DW513470
A	L	CALCULATE AND		DW513480
STO	ADD7	STORE LOW-ORDER ADD-		DW513490
STO	ADD8	RESS - CHARACTER COUNT +1.		DW513500
LDX 2	1	SET XR2 TO 1.		DW513510
LIRF	ZIPCO	CONVERT FROM EBCDIC TO		DW513520
DC	/0101	ASCII-8 ITT CODE. BOTH IN-		DW513530
ADD7 DC	**	PUT AND OUTPUT ARE UNPACK-		DW513540
ADD8 DC	**	ED. SPECIAL CODE TABLE		DW513550
ADD9 DC	**	PROVIDED BY DW057. I/O		DW513560
CALL	DW057	ADDRESS AND CHAR CNT CALC.		DW513570


```
OR MASK2
* STO STOP
LD I1 1
SRA 8
ADD02 STO L *-
*
SAVE5 LDX L1 *-
LDD L SAVEA
LDS5 LDS
MDM DW055,+3
BSCI BSC I DW055
*
* * * * * DATA AND CONSTANTS AREA * * * * *
*
DIS DC /012C
*
LIBF LIBF PAPT
MASK1 DC /FF00
MASK2 DC /0040
*****
BUF DC *-
BSS 80
*****
END
// DUP
*DELET
*STORE
WS UA
DW051
DW051
```

TOTAL CARDS LISTED 47

E.2

Card placement for punching tape.

// Job

Source
Program

// XEQ

Data
Cards

1st card - "J"s punched in column 1-14
Object Deck
Last card - "L"s punched in column 1-14

APPENDIX F

μ - MITE DESCRIPTION OF ASSEMBLY LANGUAGE FOR THE PROGRAMMABLE CONTROLLER

By: T. E. Trebelhorn

F-1

MNEMONIC OP CODE ASSEMBLER

- A - 1 Mnemonic OP Code Summary
- A - 3 Mnemonic OP Code Modifiers

SYMBOLIC COMMAND FORMATS

- B - 1 TYPE 1 (AND, ADD, INC, TC, SUB)
- B - 3 TYPE 2 (SLL, SLA, SRL, SRA)
- B - 4 TYPE 3 (LD, ST)
- B - 5 TYPE 4 (LL, LP)
- B - 6 TYPE 5 (BI, BC, BN, BO, BP, B1, B2, BZ)
(BI/, BC/, BN/, BO/, BP/, B1/, B2/, BZ/)
- B - 7 TYPE 6 (J)
- B - 8 TYPE 7 (LK)
- B - 9 TYPE 8 (CNT)
- B - 9 TYPE 9 (W, NOP)
- B - 10 TYPE 10 (EOR, OR)
- B - 10 TYPE 11 (XI/, XC/, XN/, XO/, XP/, X1/, X2/, XZ/)
- B - 11 CARD FORMAT EXAMPLES
- B - 12 μ -MITE PSEUDO-OPERATION DESCRIPTIONS
- B - 13 PSEUDO-OP CARD FORMAT EXAMPLES

μ -MITE COMMAND DESCRIPTIONS

- C - 1 ADD
- C - 2 AND
- C - 3 BRANCH ON CONDITION (NOT) - BS (BS/)
- C - 4 COUNT - CNT
- C - 5 COMPARE ACCUMULATOR - CPA

- C - 6 EXCLUSIVE OR - EOR
- C - 7 INCREMENT - INC
- C - 8 JUMP - J
- C - 9 LOAD - LD
- C - 10 LINK - LK
- C - 11 LOAD LITERAL - LL
- C - 12 OR
- C - 13 SHIFT LEFT ARITHMETIC - SLA
- C - 14 SHIFT LEFT LOGICAL - SLL
- C - 15 SHIFT RIGHT ARITHMETIC - SRA
- C - 16 SHIFT RIGHT LOGICAL - SRL
- C - 17 STORE - ST
- C - 18 SUBTRACT - SUB
- C - 19 TWO'S COMPLEMENT - TC
- C - 20 BRANCH AFTER EXECUTE - XS/

SOFTWARE IMPLEMENTED COMMANDS

- C - 21 LOAD PAGE NUMBER - LP
- C - 22 NO-OPERATION - NOP
- C - 23 WAIT - W

- D - 1 μ-MITE ERROR CODE DESCRIPTIONS

ANEMONIC	OP CODE	INSTRUCTION	CARD FORMAT TYPE	OP CODE MODIFIERS
ADD	40	ADD (OPERAND) TO (ACC) $\rightarrow$ (ACC)	1	RZ, M1, M2
AND	00	AND (OPERAND) TO (ACC) $\rightarrow$ (ACC)	1	RZ, M1, M2
B1	C6	BRANCH IF SENSE SW 1 ON	5	M4
B1/	D6	BRANCH IF SENSE SW 1 OFF	5	M4
B2	C7	BRANCH IF SENSE SW 2 ON	5	M4
B2/	D7	BRANCH IF SENSE SW 2 OFF	5	M4
BC	C3	BRANCH IF CARRY BIT ON	5	M4
BC/	D3	BRANCH IF CARRY BIT OFF	5	M4
BI	C5	BRANCH IF INTERRUPTS ACTIVE	5	M4
BI/	D5	BRANCH IF INTERRUPTS NOT ACTIVE	5	M4
BN	C1	BRANCH IF NEG FLAG ON	5	M4
BN/	D1	BRANCH IF NEG FLAG OFF	5	M4
BO	C4	BRANCH IF OVERFLOW FLAG ON	5	M4
BO/	D4	BRANCH IF OVERFLOW FLAG OFF	5	M4
BP	C0	BRANCH IF POS FLAG ON	5	M4
BP/	D0	BRANCH IF POS FLAG OFF	5	M4
BZ	C2	BRANCH IF ZERO FLAG ON	5	M4
BZ/	D2	BRANCH IF ZERO FLAG OFF	5	M4
CNT	B0	INCREMENT (REG) BRANCH DIRECTLY IF (REG) $\neq$ 0	8	RY
CPA	20	COMPARE (ACC) TO (OPERAND)	1	RZ, M1, M2
EOR	14	EXCLUSIVE OR (OPERAND) TO (ACC) $\rightarrow$ (ACC)	10	RZ, M3
INC	50	INCREMENT (OPERAND) + 1 $\rightarrow$ (ACC)	1	RZ, M1, M2
J	E0	JUMP TO OPERAND OPERAND $\rightarrow$ (RE) (RL)	6	
LD	80	LOAD (OPERAND) $\rightarrow$ (REG)	3	RY, M3
LK	F0	LINK (RL) $\rightarrow$ (REG), (OPERAND) $\rightarrow$ (RL)	7	RY, M5
LL	90	LOAD LITERAL OPERAND $\rightarrow$ (REG)	4	RY, M3
LP	90	LOAD OPERAND PAGE OPERAND PAGE NO. $\rightarrow$ (REG)	4	RY, M3

MNEMONIC	OP CODE	DESCRIPTION	CARD FORMAT TYPE	OP CODE MODIFIERS
NOP	80	NO OPERATION	9	
OR	10	OR (OPERAND) TO (ACC) → (ACC)	10	RZ, M3
SLA	30	SHIFT LEFT ARITHMETIC (OPERAND) → (ACC) SHIFT 1 BIT LA	2	RZ
SLL	38	SHIFT LEFT LOGICAL (OPERAND) → (ACC) SHIFT 1 BIT LL	2	RZ
SRA	34	SHIFT RIGHT ARITHMETIC (OPERAND) → (ACC) SHIFT 1 BIT RA	2	RZ
SRL	3C	SHIFT RIGHT LOGICAL (OPERAND) → (ACC) SHIFT 1 BIT RL	2	RZ
ST	A0	STORE (REG) → (OPERAND)	3	RY, M3
SUB	70	SUBTRACT (OPERAND) FROM (ACC) → (ACC) 1		RZ, M1, M2
TC	60	TWO'S COMPLEMENT T.C. (OPERAND) → (ACC)	1	RZ, M1, M2
W	E (*)	WAIT	9	
X1/	3E	BRANCH AFTER EXECUTE S.S. 1 OFF	11	
X2/	BF	BRANCH AFTER EXECUTE S.S. 2 OFF	11	
XC/	BB	BRANCH AFTER EXECUTE CARRY FL. OFF	11	
XI/	BD	BRANCH AFTER EXECUTE INT FL. OFF	11	
XN/	B9	BRANCH AFTER EXECUTE NEG FL. OFF	11	
XO/	BC	BRANCH AFTER EXECUTE OVERFLOW FL. OFF	11	
XP/	B8	BRANCH AFTER EXECUTE POS. FL. OFF	11	
XZ/	BA	BRANCH AFTER EXECUTE ZERO FL. OFF	11	

OP CODE MODIFIERS

<u>TYPE</u>	<u>MNEMONIC</u>	<u>DESCRIPTION</u>	<u>OP CODE MODIFIERS</u>
RY	RA	A - Register	00
	RB	B - Register	01
	RC	C - Register	02
	RD	D - Register	03
	RE	Extend Register	04
	RL	L - Register	05
	RX	Index Register	06
	RI	Interrupt Register	07
RZ	RA	A - Register	00
	RB	B - Register	01
	RC	C - Register	02
	RD	D - Register	03
M1	X	Index Mode	08
	MP	Multiprecision Mode	04
	(b)	Blank (No Modifier)	00
M2	X	Index Mode	08
	MP	Multiprecision Mode	04
	(b)	Blank (No Modifier)	00
M3	X	Index Mode	08
	(b)	Blank (No Modifier)	00
M4	I	Indirect Branch	00
	(b)	Blank (No Modifier)	08
M5	D	Disable Interrupt	08
	(b)	Blank (No Modifier)	00

MICROMITE SYMBOLIC COMMAND FORMATS

TYPE 1 (See Example A)

Col 8 - 13 Six character symbolic tag used to define a location in control memory

Col 15-17 Symbolic Operation Code

AND	And
ADD	Add
INC	Increment
TC	Two's Complement
SUB	Subtract

Col 20-29 OPERAND

1.) One-six character symbolic tag used to define the word of storage memory to be used. If the OPERAND specified is not within the range of directly addressable storage memory, a page register must also be specified. This is done by placing a 0, 1, or 2 in parenthesis immediately following the symbolic tag.

2.) Two hex character literal to specify the value to be placed in the "D" field of this command.

All hex literals use Col. 28 - 29.

Col 31-39 MODIFIERS

RZ Register modifier used to define which register this command will operate on.

Acceptable registers are:

RA	A - Register
RB	B - Register
RC	C - Register
RD	D - Register

This modifier must always be used.

M1 Command Modifier One

Three possibilities for M1 exist within the type 1 command code group

X Index mode to be used

MP Multiprecision mode

b If blank, no M1 or M2 is to be used.

M2 Command Modifier Two

The same three modifiers used in M1 are also legal for use in M2.

X Index mode

MP Multiprecision mode

b If blank, no M2 is to be used.

All modifiers must be separated by commas.

Col 41 - 72

Comments

TYPE 2 (See Example B)

Col 8 - 13 (Same as for Type 1)

Col 15 - 17 Symbolic Operation Code

SLL Shift Left Logical
 SLA Shift Left Arithmetic
 SRL Shift Right Logical
 SRA Shift Right Arithmetic

Col 20 - 29 OPERAND (Same as for Type 1)

Col 31 - 39 Modifiers

RZ Register Modifier used to define which register
 this command will operate on.

Acceptable registers are:

RA A - Register
 RB B - Register
 RC C - Register
 RD D - Register

This modifier must always be used.

No other modifier is valid for Type 2 OP Codes 1 .

Col 41 - 72 Comments

TYPE 3

(See Example C)

Col 8 - 13

(Same as for Type 1)

Col 15 - 17

Symbolic Operation Code

LD Load

ST Store

Col 20 - 29

OPERAND (Same as for Type 1)

Col 31 - 39

Modifiers

RY Register modifier used to define which register this command will operate on.

Acceptable Registers are:

RA A - Register

RB B - Register

RC C - Register

RD D - Register

RE Extend Register

RL L - Register

RX Index Register

RI Interrupt Register

This modifier must always be used.

M3 Command modifier three

Two possibilities for M3 exist within the Type 3 command code group.

X Index Mode

b If blank, no modifier

Col 41 - 72

Comments

TYPE 4

(See Example D)

Col 8 - 13

Ref. Label (Same as Type 1)

Col 15 - 17

Symbolic Operation Code

LL Load Literal

LP Load Page

Col 20 - 29

OPERAND

- 1.) One-six character symbolic tag used to specify the address (or page) of control or storage memory to be placed in the "D" field of this command.
- 2.) Two hex character literal to specify the value to be placed in the "D" field of this command.

All hex literals use Col 28 - 29.

Col 31 - 39

Modifiers (Same as Type 3)

Col 41 - 72

Comments

TYPE 5

(See Example E)

Col 8 - 13

Ref Label (Same as Type 1)

Col 15 - 17

Symbolic Operation Code

BI	(BI/)	BR	Interrupt Active (Not)
BC	(BC/)	BR	Carry Set (Not)
BN	(BN/)	BR	Neg (Not)
BO	(BO/)	BR	Overflow Flag (Not)
BP	(BP/)	BR	Positive (Not)
B1	(B1/)	BR	Sense Switch 1 (Not)
B2	(B2/)	BR	Sense Switch 2 (Not)
BZ	(BZ/)	BR	Zero (Not)

Col 20 - 29

OPERAND

- 1.) One-six character symbolic tag used to specify the address of control memory to be placed in the "D" field of this command if the indirect modifier is not set.
- 2.) One-six character symbolic tag used to specify the address of storage memory to be placed in the "D" field of this command if the indirect modifier is set. If a page register is used, it must be specified by placing a 0, 1, or 2 in parenthesis immediately following the symbolic tag.
- 3.) Two hex character literal to specify the value to be placed in the "D" field of this command.

All hex. literals use Col 28 - 29.

Col 31 - 39

Modifiers

M4 Command Modifier Four

Two possibilities for M4 exist within the type 5 command code group.

I	Indirect branch address
b	Direct branch address

Col 41 - 72

Comments

TYPE 6

(See Example F)

Col 8 - 13

Ref Label (Same as Type 1)

Col 15 - 17

Symbolic Operation Code

J Jump (Unconditional)

Col 20 - 29

OPERAND

- 1.) One - six character symbolic tag used to specify the absolute control memory address to be used by this command.
- 2.) Three character hex literal used to specify the absolute control memory address to be used by this command. This literal is placed in Col 27 - 29.

Col 41 - 72

Comments

TYPE 7

(See Example G)

Col 8 - 13

Ref Label (Same as Type 1)

Col 15 - 17

Symbolic Operation Code

LK Link

Col 20 - 29

OPERAND

- 1.) One-six character symbolic tag used to define the address of control memory to be placed in the "D" field of this command.
- 2.) Two character hex literal used to specify the value to be placed in the "D" field of this command.

Two character hex literals use Col 28 - 29.

Col 31 - 39

Modifiers

RY Register Modifier used to define which register this command will use.

Acceptable Registers are:

RA	A	- Register
RB	B	- Register
RC	C	- Register
RD	D	- Register
RE	E	- Extend Register
RL	L	- Register
RX		Index Register
RI		Interrupt Register

M5 Command Modifier Five

D	Disable Interrupt
b	Blank, No Modifier

Col 41 - 72

Comments

TYPE 8

(See Example H)

Col 8 - 13	Ref Label (Same as Type 1)
Col 15 - 17	Symbolic Operation Code
	CNT Count
Col 20 - 29	OPERAND (Same as Type 7)
Col 31 - 39	Modifiers
	RY Register Modifier used to define which register this command will operate on.
	Acceptable registers are:
	RA A - Register
	RB B - Register
	RC C - Register
	RD D - Register
	RE Extend Register
	RL L - Register
	RX Index Register
	RI Interrupt Register

This Modifier must always be used.No other modifier is valid for Type 8 OP Codes.

Col 41 - 72

Comments

TYPE 9

(See Example I)

Col 8 - 13	Ref Label (Same as Type 1)
Col 15 - 17	Symbolic Operation Code
	W Wait
	NOP No Operation
Col 20 - 29	OPERAND (none)
Col 31 - 39	Modifiers (none)
Col 41 - 72	Comments

TYPE 10

(See Example J)

Col 8 - 12

Ref Label (Same as Type 1)

Col 15 - 17

Symbolic Operation Code

EOR Exclusive Or

OR OR

Col 20 - 29

OPERAND (Same as Type 1)

Col 31 - 39

Modifiers

RZ Register modifier used to define which register this command will operate on.

Acceptable Registers are:

RA A - Register

RB B - Register

RC C - Register

RD D - Register

This Modifier must always be used.

M3 Command Modifier Three

X Index Mode

b If blank, no modifier

Col 41 - 72

Comments

TYPE 11

(See Example K)

Col 8 - 13

Ref Label (Same as Type 1)

Col 15 - 17

Symbolic Operation Code

X1/ BR. AFTER EXECUTE S.S.W. 1 OFF

X2/ BR. AFTER EXECUTE S.S.W. 2 OFF

XC/ BR. AFTER EXECUTE CARRY FL. OFF

XI/ BR. AFTER EXECUTE INTERRUPT NOT ACTIVE

XN/ BR. AFTER EXECUTE OVERFLOW FL. OFF

XO/ BR. AFTER EXECUTE OVERFLOW FL. OFF

XP/ BR. AFTER EXECUTE NOT POS.

XZ/ BR. AFTER EXECUTE ZERO FL. OFF

Col 26 - 29

OPERAND (Same as Type 7)

Col 31 - 39

Modifiers (None)

Col 41 - 72

Comments

ASSEMBLY LANGUAGE PSEUDO - OPERATIONS

DATA DEFINITIONS

BCI	Insert the binary equivalence to the alpha-numeric characters in the comment field (Col 41-72) into the first available contiguous single page block of length equal to the decimal length of the field as defined in Col 27 - 29 of the OPERAND field. (See Example 1)
BSS	Block Save - The number of contiguous single page memory locations (max. 64); as specified by the decimal value contained in Col 27 - 29 of the OPERAND field. (See Example 2)
DS	Define Storage - Save one memory location. If a specific hex page-displacement is defined in Col 37 - 39 of the modifier field, the storage location will be defined at that location. If Col 37 - 39 are left blank, the next available memory location will be used. (See example 3)
DA	Define Address Displacement. The displacement of the ref tag defined in the OPERAND (Col 20-26) is stored in this memory location. Address specification is the same as DS. (See Example 4)
DEC	Insert the binary equivalent to the decimal value specified in Col 27-29 of the OPERAND field. Address specification is the same as DS. (See Example 5)
EQU	Use the same storage address for the term specified in the reference field (Col 8-13) as the previously defined term listed in the OPERAND field (Col 20-26). (See Example 6)
HEX	Insert the hex value specified in Col 28-29 of the OPERAND field. Address specification is the same as DS. (See Example 7)

ASSEMBLER CONTROL

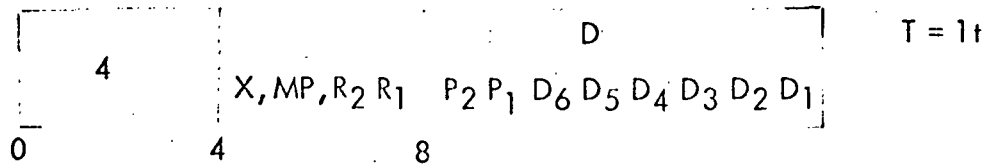
ORG	Origin - Set program counter to the hex value specified in Col 27-29 of the OPERAND field. (See Example 8)
END	End of Assembly

F - MITE COMMAND DESCRIPTIONS

ABBREVIATIONS

0	Value in this Bit Position is Zero
1	Bit Value Position One Contains a One
2	Bit Value Position Two Contains a One
4	Bit Value Position Four Contains a One
8	Bit Value Position Eight Contains a One
$S_{3, 2, 1}$	Branch Condition Modifier
D	D Field
D_{6-1}	Displacement for Storage Memory Use
D_{8-1}	Displacement for Control Memory Use
I	1) Interrupt Inhibit Modifier 2) Instruction Counter
I_c	Present Value of Instruction Counter
L	Indirect Branch Modifier
MP	Multi-Precision Modifier
$P_{2, 1}$	Page Register Specification Bits
t	Machine Cycle (≈ 200 nSec)
T	Command Timing
X	Index Modifier

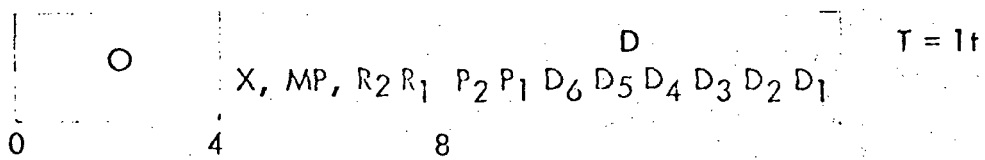
ADD



The contents of the location specified in the D Field is ADDED to the contents of the Accumulator specified by R₂ R₁. If the Index Modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the Page Displacement (D₆ - D₁), before command execution. If the Multi-Precision Modifier is set, the carry bit is added with both operands, enabling wide word arithmetic capability.

ADD (OPERAND) TO (ACC) → (ACC)

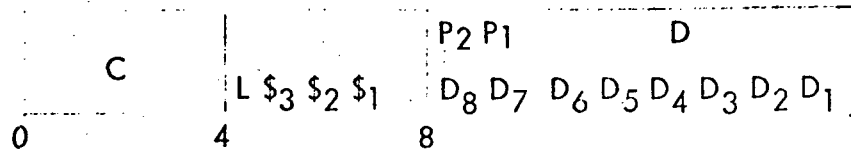
AND



The contents of the Accumulator specified by R₂ R₁ is ANDED with the contents of the location specified in the D Field. If the Index Modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the Page Displacement (D₆ - D₁), before command execution.

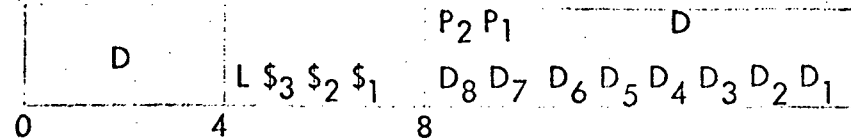
AND (OPERAND) TO (ACC) -- (ACC)

B\$ (BRANCH IF \$)



T = 2t

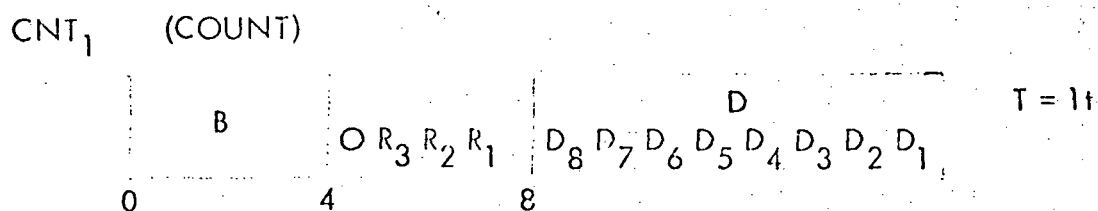
B\$/ (BRANCH IF NOT \$)



T = 2t

A branch occurs if the condition specified by $\$3 \$2 \$1$ is met. If the L Modifier is set to 0 by using the M_4 Op Code Modifier, the branch address is loaded from the Page Disp. specified in the D Field. If the M_4 Op Code Modifier is not used, the L modifier is set to 1 and the branch address is loaded directly from the D Field of the command word.

B\$	(B\$/)	$\$3$	$\$2$	$\$1$	
BP	(BP/)	0	0	0	POS
BN	(BN/)	0	0	1	NEG
BZ	(BZ/)	0	1	0	ZERO
BC	(BC/)	0	1	1	CARRY
BO	(BO/)	1	0	0	OVERFLOW
BI	(BI/)	1	0	1	INTERRUPTS ACTIVE
B1	(B1/)	1	1	0	S.S. 1
B2	(B2/)	1	1	1	S.S. 2

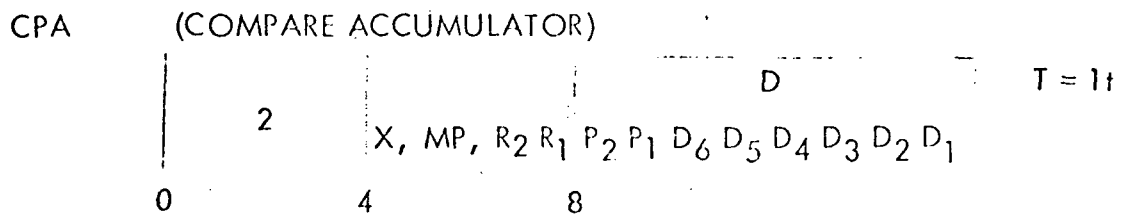


The contents of the Special Register specified by $R_3 R_2 R_1$ is incremented by 1 and a direct branch to D is taken at $I = I_C + 2t$ if the new value contained in the Special Register is $\neq 0$. The instruction immediately following the CNT command is always executed regardless of the branch condition which occurs at $I = I_C + 2t$.

$(REG) + 1 \rightarrow (REG)$

Branch directly if $(REG) \neq 0$

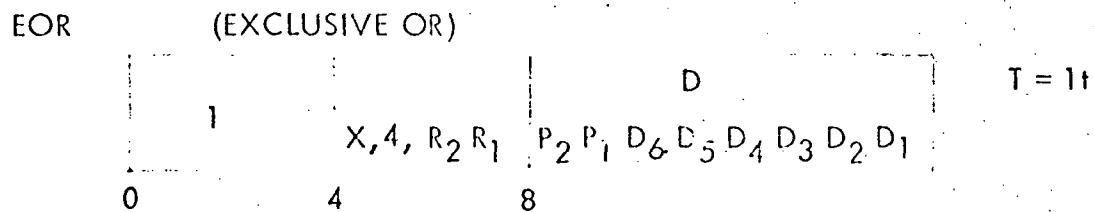
- 1 Execute & Branch Instruction (i.e., Instruction at $I = I_C + 1t$ is always executed)



Execution of the Compare Accumulator command causes the machine to subtract the contents of the location specified in the D Field from the contents of the accumulator specified by R₂ R₁ in such a way that the arithmetic flags are set but the contents of the accumulator and operand remain unaffected. If the Index modifier is set, the content of the Index Register is SINGLE PRECISION ADDED to the page displacement (D₆ - D₁) before command execution. If the Multi-Precision modifier is set, the zero flag can be used to indicate compare for wide word capability.

SUBTRACT (OPERAND) FROM (ACC)

SET ARITHMETIC FLAGS

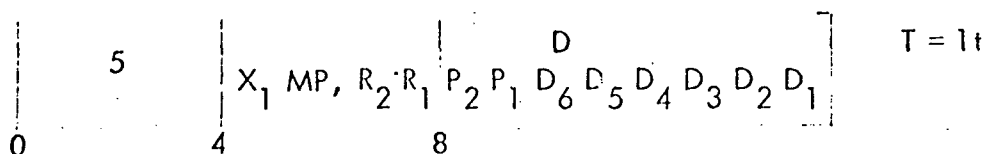


The contents of the accumulator referenced by $R_2 R_1$ is exclusive "OR"ed with the contents of the location specified in the $\underline{D}$ Field. If the Index modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the page displacement ($D_6 - D_1$) before command execution.

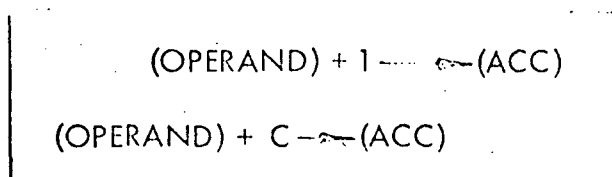
EXCLUSIVE OR

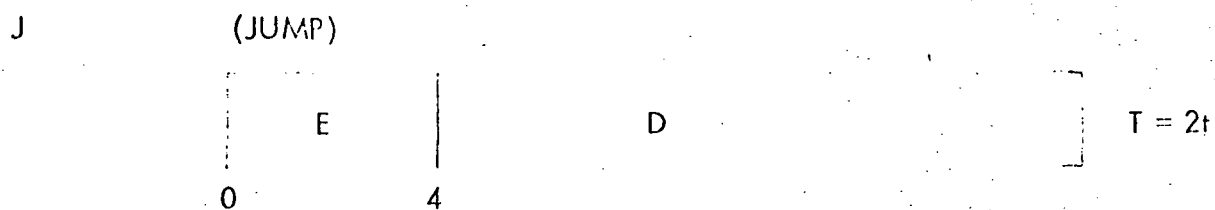
(OPERAND) TO (ACC) — (ACC)

INC (INCREMENT ACCUMULATOR)

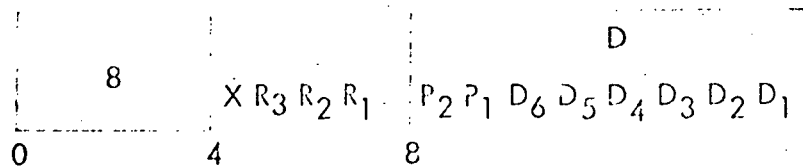


The contents of the location specified in the D Field is incremented by one and placed in the accumulator specified by $R_2 R_1$. If the Index Modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the Page Displacement ($D_6 - D_1$) before command execution. If the multi-precision modifier is set, the carry bit is added to the contents of the addressed accumulator and the increment is inhibited.



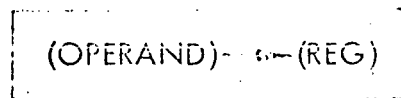


The contents of the D Field is loaded into the L Register and the lower half of the Extend Register. This forces an unconditional Jump to any of the 4096 locations in control store. The condition flags are not disturbed.

LD_{1,2} (LOAD)

T = 11

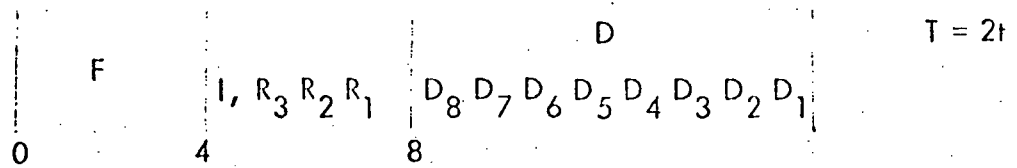
The contents of the location specified in the D field is loaded into the special register specified by $R_3 R_2 R_1$. If the Index modifier is set, the contents of the index Register is SINGLE PRECISION ADDED to the Page Displacement ($D_6 - D_1$) before command execution.



1. When the L Register is specified by $R_3 R_2 R_1$, a command execution delay is caused. (i.e. $T = 2t$).
2. When the L Register is specified by $R_3 R_2 R_1$ and the Interrupt Register is specified in the D Field, the interrupts are enabled.

LK

(LINK)



The contents of the L Register is stored in the special register specified by $R_3 R_2 R_1$. The contents of D Field in the command word is placed in the L Register. If the interrupt disable modifier is set to 1, interrupts are inhibited.

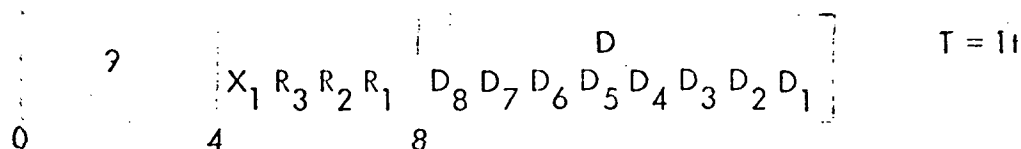
This command causes a direct branch to the location specified in the 'D' Field while saving a return address in the special register specified by $R_3 R_2 R_1$.

(RL) $\rightarrow$ (REG)

OPERAND $\rightarrow$ (RL)

LL₁

(LOAD LITERAL)

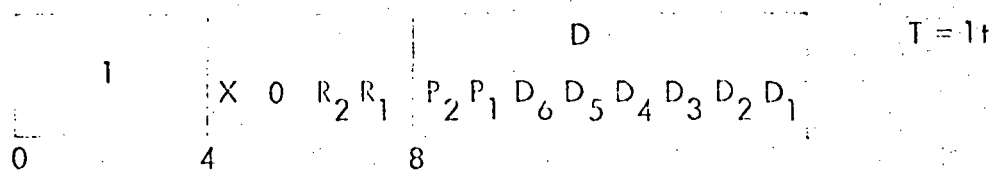


The contents of the D Field in the command word is Loaded into the special register specified by $R_3 R_2 R_1$. If the Index Modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the D Field of the command word before the execution of the instruction.

OPERAND $\rightarrow$ (REG)

1. When the L Register is specified by $R_3 R_2 R_1$, a common execution delay is caused, (i.e., $T = 2t$).

OR

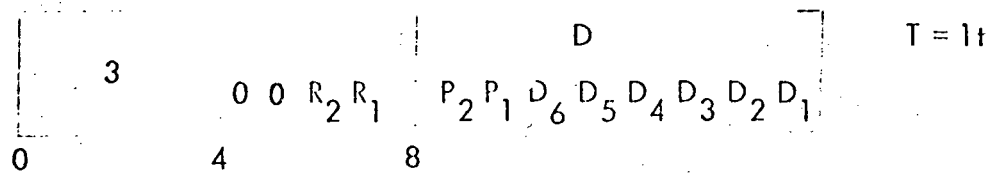


The contents of the accumulator referenced by $R_2 R_1$ is "OR"ed with the contents of the location specified in the D Field. If the Index modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the Page Displacement ($D_6 - D_1$) before command execution.

OR
(OPERAND) TO (ACC) $\leftarrow$ (ACC)

SLA

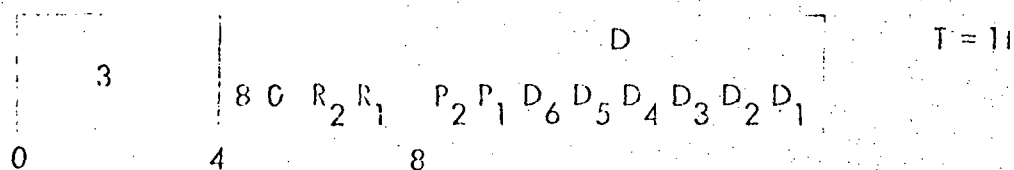
(SHIFT LEFT ARITHMETIC)



The contents of the location specified in the D Field is loaded into the accumulator specified by $R_2 R_1$. The accumulator is then left shifted 1 bit position with the right most bit position being zero filled.

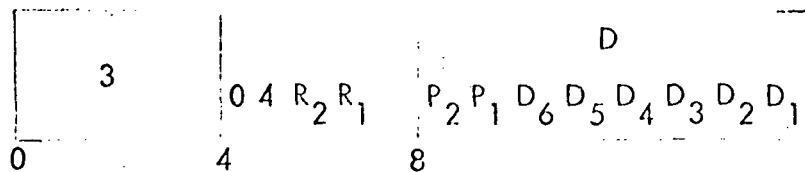
SLI.

(SHIFT LEFT LOGICAL)



The contents of the location specified in the D Field is loaded into the accumulator specified by R₂ R₁. The accumulator is then left shifted 1 bit position with the contents of the carry bit register being shifted into the right-most bit position. The bit shifted out of the accumulator is shifted into the Carry bit Register enabling wide word shifts.

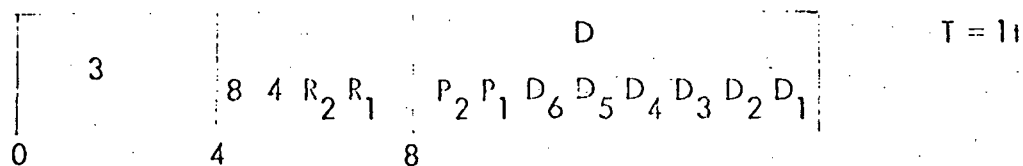
SRA. (SHIFT RIGHT ARITHMETIC)



T = 1t

The contents of the location specified in the D Field is loaded into the accumulator specified by $R_2 R_1$. The accumulator is then right shifted 1 bit position with the left most bit retaining the value of the sign bit.

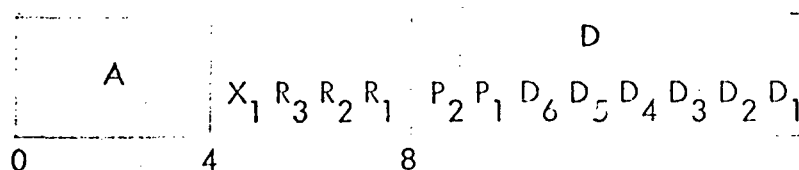
SRL (SHIFT RIGHT LOGICAL)



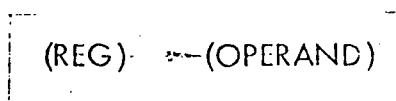
The contents of the location specified in the D Field is loaded into the accumulator specified by R₂ R₁. The accumulator is then shifted right 1 bit position with the contents of the Carry Bit Register being shifted into the left most bit position. The bit shifted out of the accumulator is shifted into the Carry Bit Register enabling wide word shifts.

ST

(STORE)

 $T = 1t$

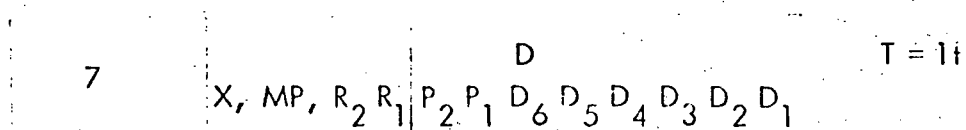
The contents of the special register addressed by $R_3 R_2 R_1$ is stored in the location specified in the D Field. If the Index Modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the Page Displacement ($D_6 - D_1$) before command execution.



1. If the L Register is specified by the D Field, a command execution delay will be caused. (i.e., $T = 2t$)

SUB

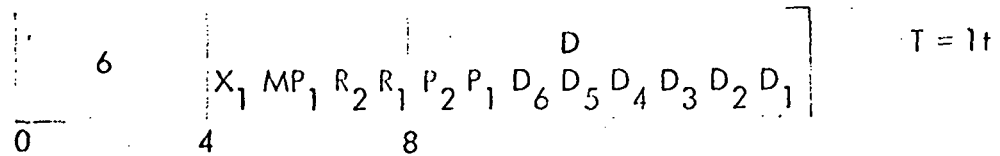
(SUBTRACT)



The contents of the location specified in the D Field is Subtracted from the contents of the accumulator specified by R₂ R₁. If the Index Modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the Page Displacement (D₆ - D₁) before command execution. If the Multi-Precision Modifier is set, the carry bit is added to the result.

SUBTRACT (OPERAND) FROM (ACC) -- (ACC)

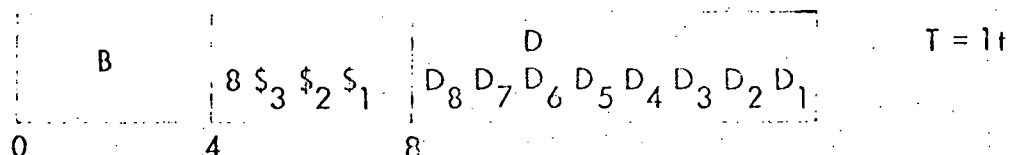
TC (TWO'S COMPLEMENT)



The contents of the location specified in the D Field is Two's Complemented and placed in the accumulator specified by R₂ R₁. If the Index Modifier is set, the contents of the Index Register is SINGLE PRECISION ADDED to the Page Displacement before command execution. If the multi-precision modifier is set, the zero flag can be reset, but not set and the carry bit is added without forcing it as in single precision.

TWO'S COMPLEMENT
T.C. (OPERAND) → (ACC)

XS/_{1,2} (BRANCH AFTER EXECUTE, IF NOT S)



A direct branch to the location specified in the D Field of this command will be executed after execution of the next instruction immediately following the XS/ instruction, if the conditions $S_3 S_2 S_1$ specified for the branch are met. If the conditions for the branch are not met, sequential command execution continues with no command execution delay.

XS/	S_3	S_2	S_1	
XP/	0	0	0	POS
XN/	0	0	1	NEG
XZ/	0	1	0	ZERO
XC/	0	1	1	CARRY
XO/	1	0	0	OVERFLOW
XI/	1	0	1	INTERRUPT ACTIVE
X1/	1	1	0	SS1
X2/	1	1	1	SS2

1. Execute & Branch Instruction

(i.e. Instruction at $I = I_c + 1t$ is always executed.)

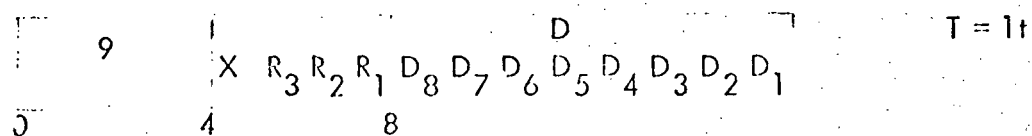
.. The branch will occur at $I = I_c + 2t$.

ASSEMBLER IMPLEMENTED OPERATIONS

LP	Load Page	LL (Ref. Page No.)
NOP	No Operation	Load (RA) $\rightarrow$ (RA)
W	Wait	J* (Jump to Same)

LP

LOAD PAGE

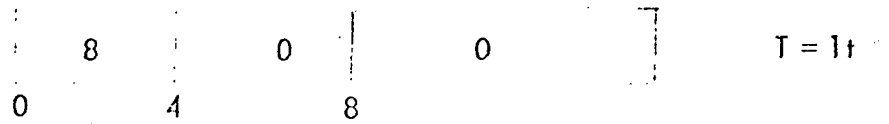


The LP operation is a software implemented command. When the LP Op-Code is recognized, the assembler generates a Load Literal command with the page number of the reference tag specified in the operand as the value contained in the D Field.

OPERAND PAGE NO. → (REG)

NOP

NO OPERATION



The NOP operation is a software implemented command. When the NOP Op-Code is recognized, the assembler generates a Load command such that location 0 of scratch pad memory is loaded into location 0 of scratch pad memory. The only effects of this operation are: 1) a one cycle delay in command execution and 2) occupy one address of control memory such that the normal operation of the program counter is not interrupted.

LOAD (LOCATION 0) → (LOCATION 0)



The Wait operation is a software implemented command. When the Wait Op Code is recognized, the assembler generates a Jump command with the D Field containing the absolute address of the "Wait" instruction. This causes the CPU to cycle on the "Wait" instruction until the program counter (L Register) is altered. (i.e. Interrupt or reset)

JUMP TO PRESENT ADDRESS
(i.e. LOOP ON THIS ADDRESS)

μ - MITE ERROR CODES

E 000 COL 7 BAD	1) Card Col. 7 must contain a "D" (for Data) or a "C" (for Command) to be a legal entry. 2) All "D" cards must precede any "C" card entry.
E 100 NO EQU TAG	1) Program can't find Ref tag specified by EQU Pseudo-Op
E 110 JAD DATA DEF	1) Illegal data definition or Pseudo-Op. Program can not recognize operation specified.
E 120 BAD ADR. DEF	1) Program can not find Ref tag specified.
E 200 BAD OP CODE	1) Illegal Op code specified.
E 300 BAD MOD AREA	1) Modifiers specified are off-col. 2) Illegal spacing in modifier area, (i.e. No blanks allowed). 3) Missing commas (i.e. modifiers are separated with commas). 4) Illegal modifier specified.
E 310 BAD REG MOD	1) Illegal register modifier specified.
E 320 BAD 2ND MOD	1) Illegal second modifier specified.
E 330 BAD 3RD MOD	1) Illegal third modifier specified.
E 340 TOO MANY MOD	1) Too many modifiers specified for this Op-Code.
E 400 BAD OPERAND	1) Illegal operand specified.
E 500 BAD PAGE REG	1) Illegal page register specified in operand field. 2) Illegal spacing in operand field.

E 600 BAD REF TAG

1) Program cannot find Ref. Tag specified in operand field.

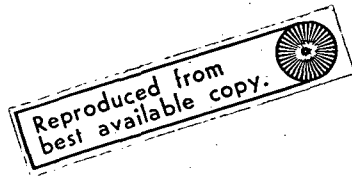
E 610 DUP REF TAG

1) This reference tag has been used at another location.

E 700 BAD LITERAL

1) Bad operand area for literal information.
2) Illegal literal for this op-code.

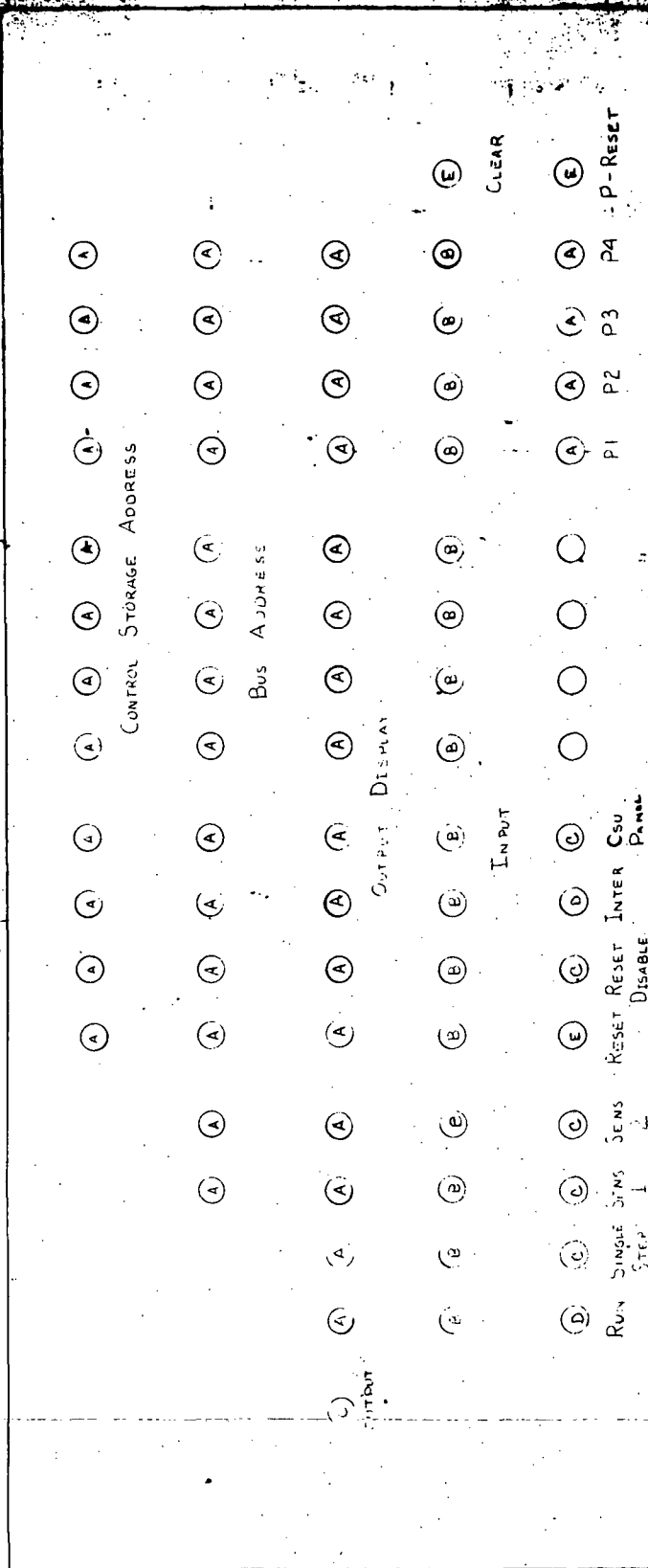
APPENDIX G



MATH MODEL PROCESSOR

AND

MAINTENANCE PANEL ELECTRONICS

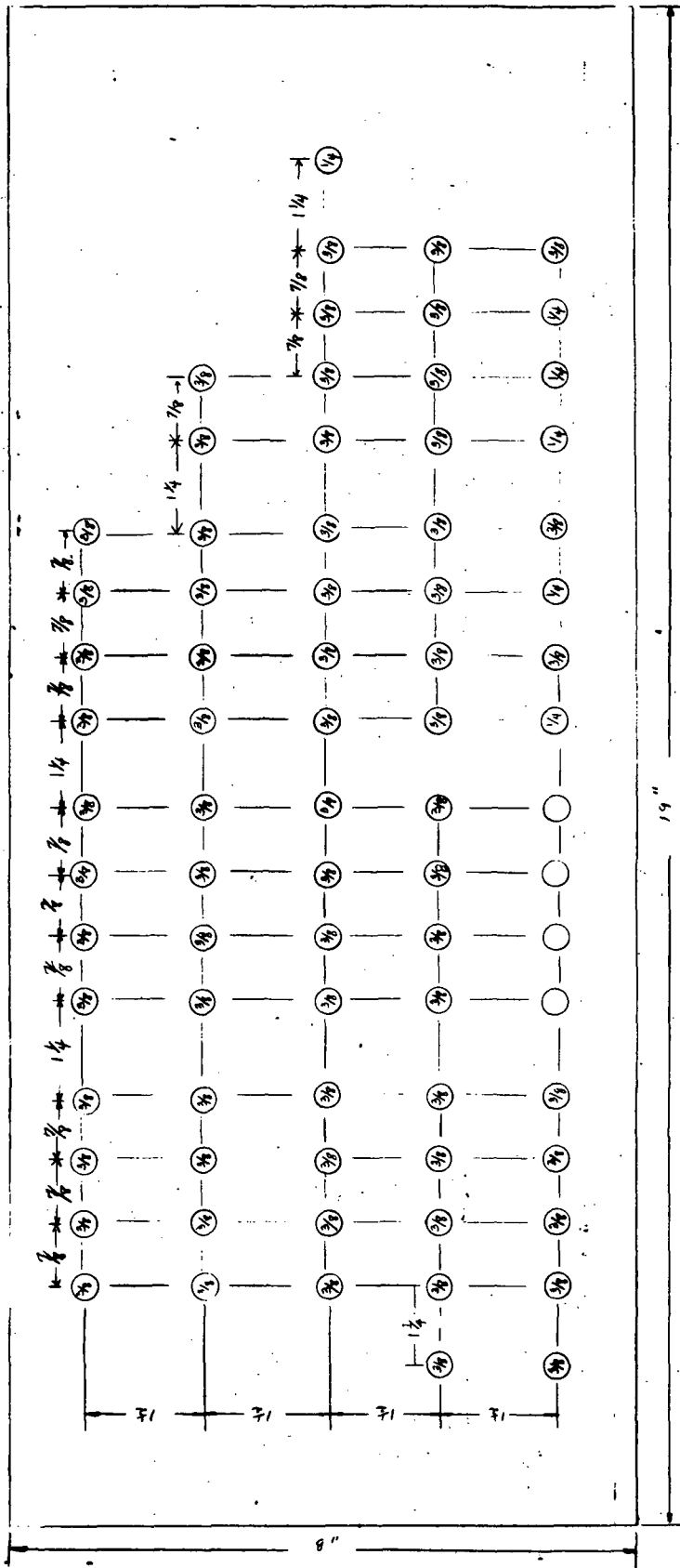


SAMPLE
A (1/2)
B (1/4)
C (1/8)
D (1/16)
E (1/32)

Front Panel Computer Requirements

REV - F

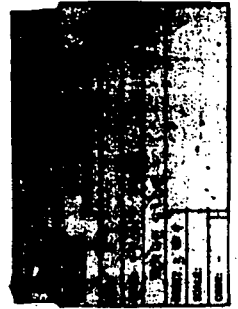




Front Panel General Layout

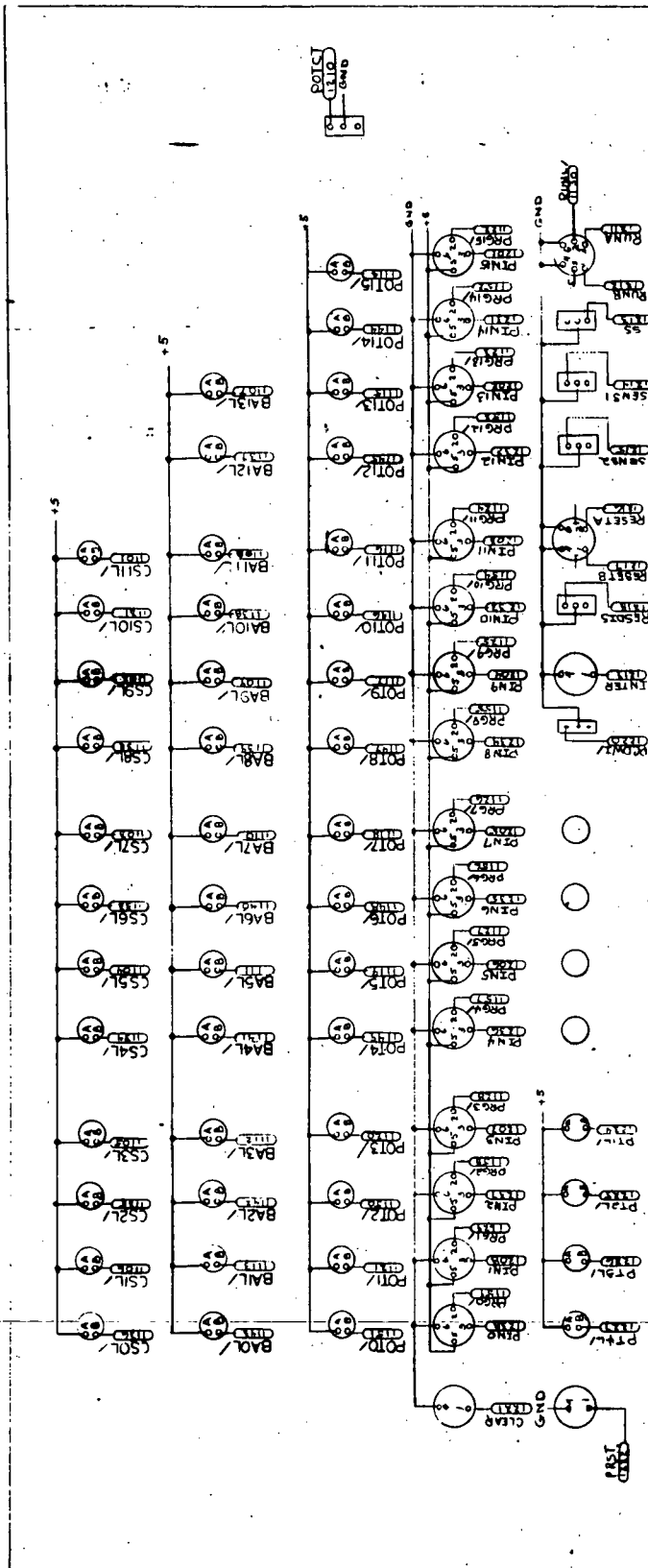
REV. =

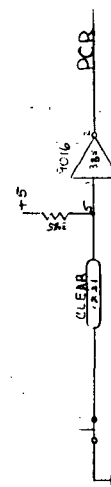
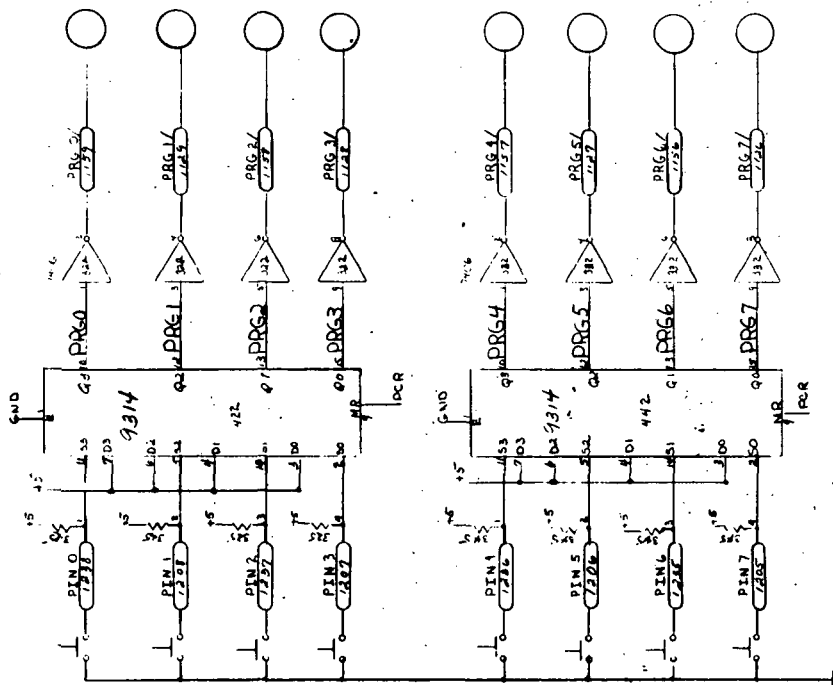
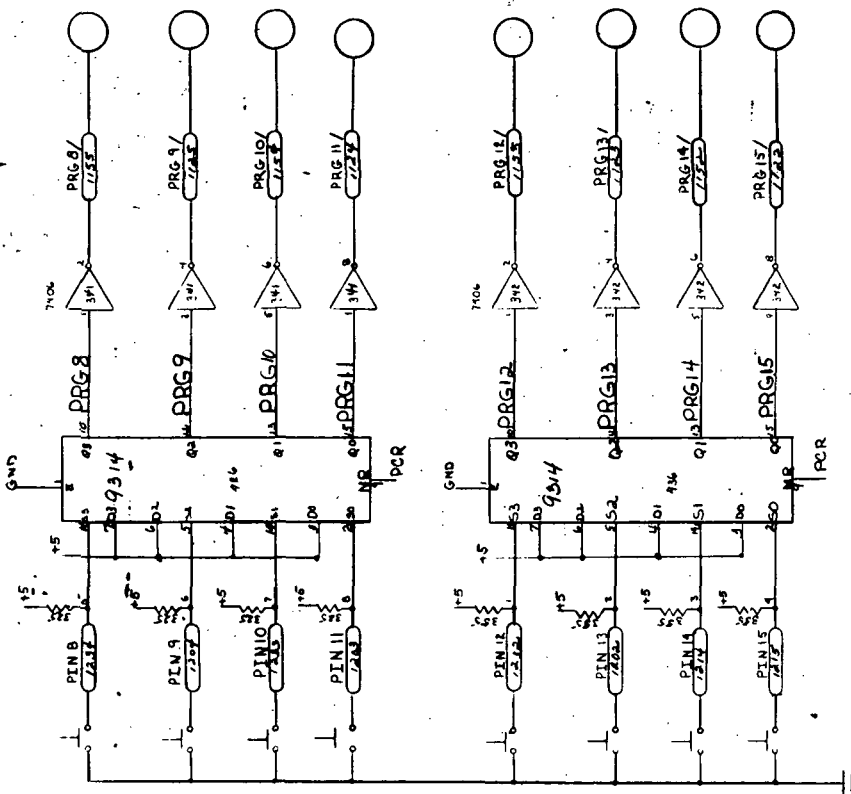
DATE	10/1/54
BY	J. W. B.
CHKD	J. W. B.
APP'D	J. W. B.
REV	1
DESCRIPTION	FRONT PANEL LAYOUT
PROJECT	REAR VIEW
DATE	10/1/54
BY	J. W. B.
CHKD	J. W. B.
APP'D	J. W. B.
REV	1
DESCRIPTION	FRONT PANEL LAYOUT
PROJECT	REAR VIEW

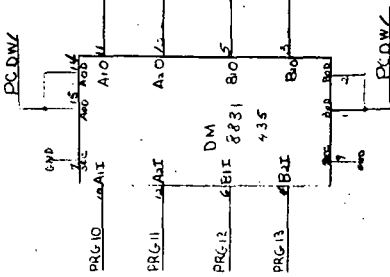
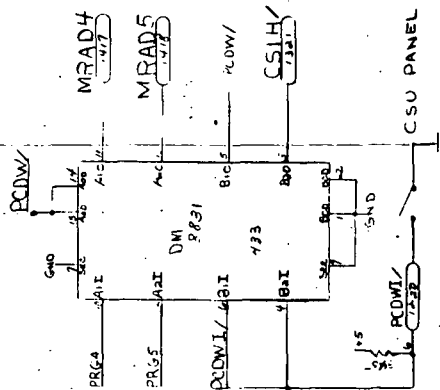
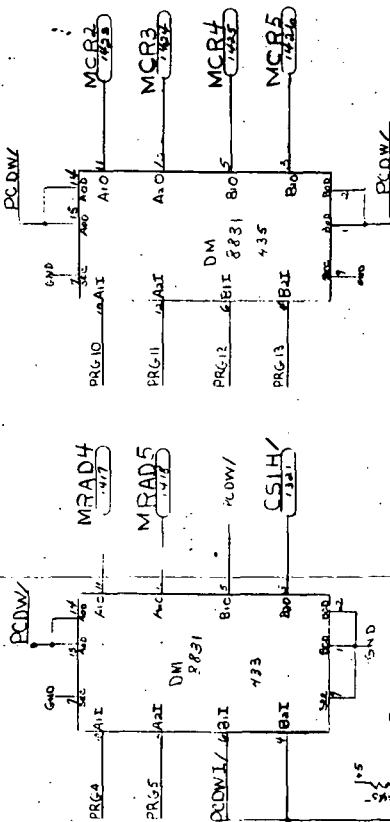
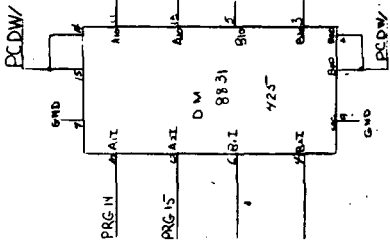
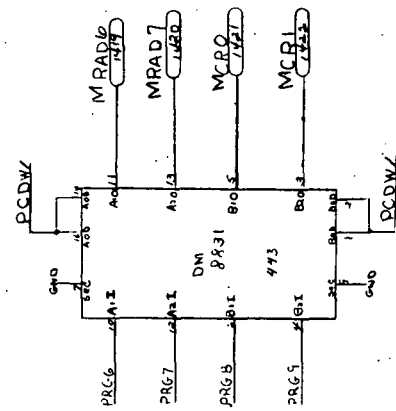
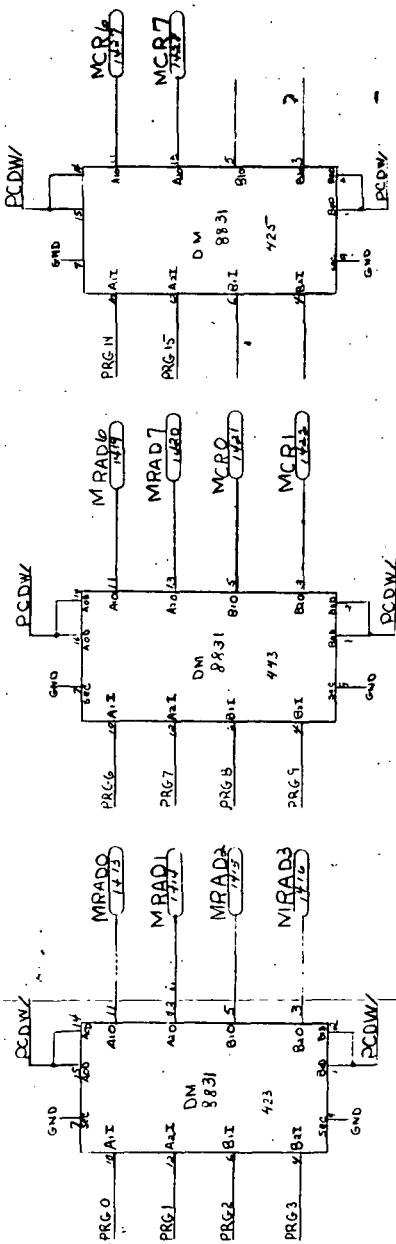


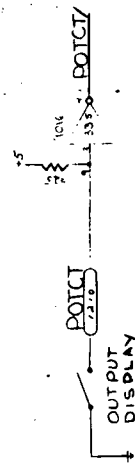
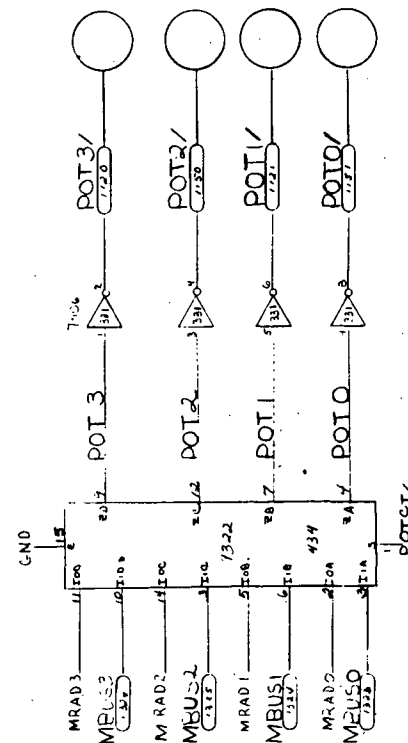
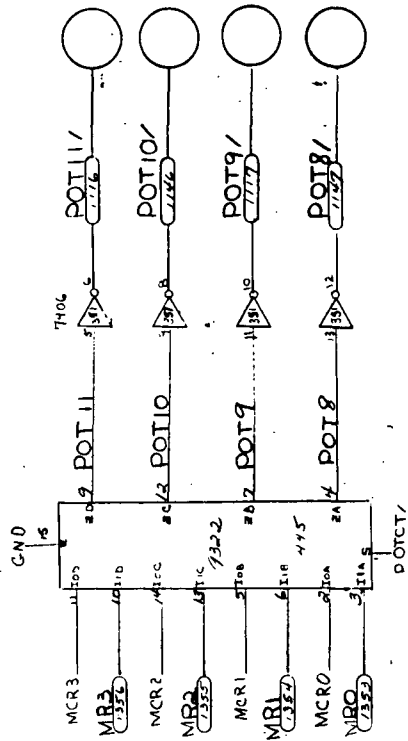
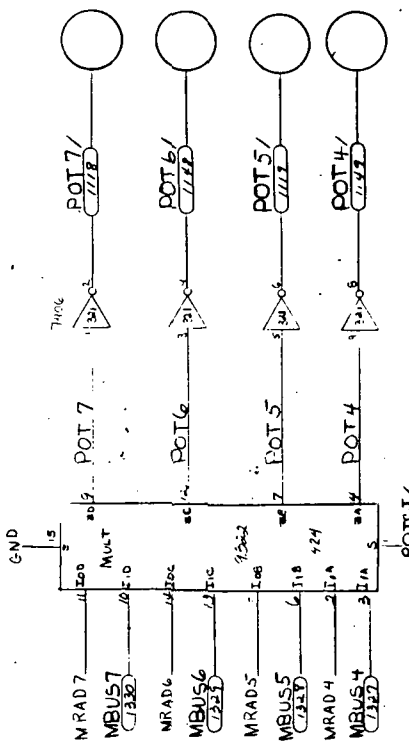
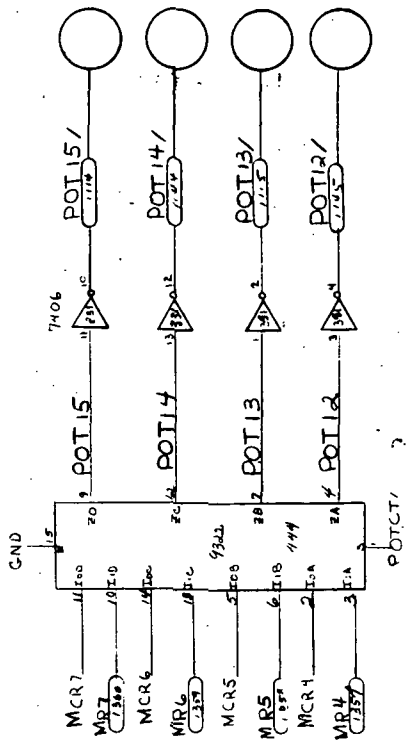
REV.-F

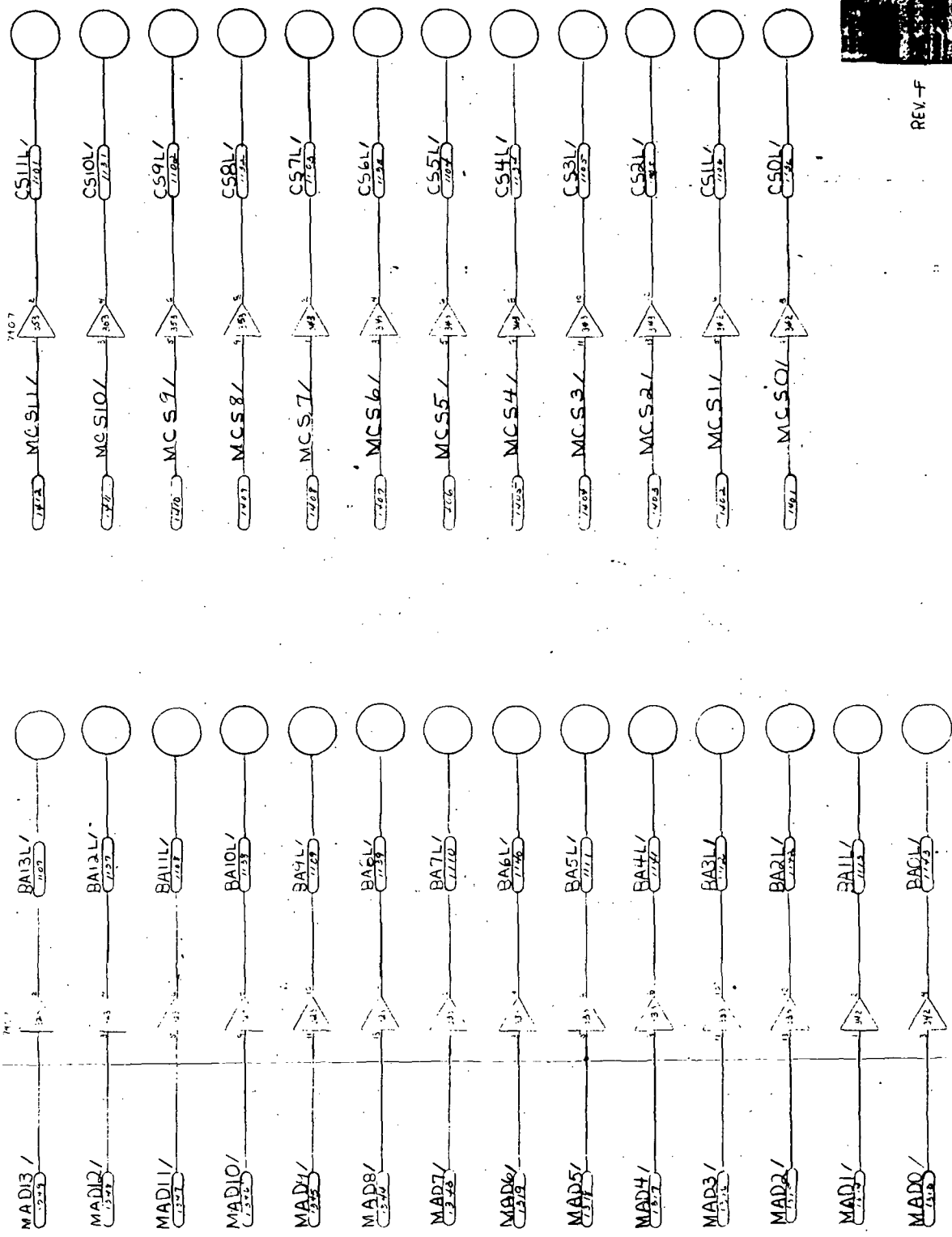
Front Panel Wire Side Interconnects

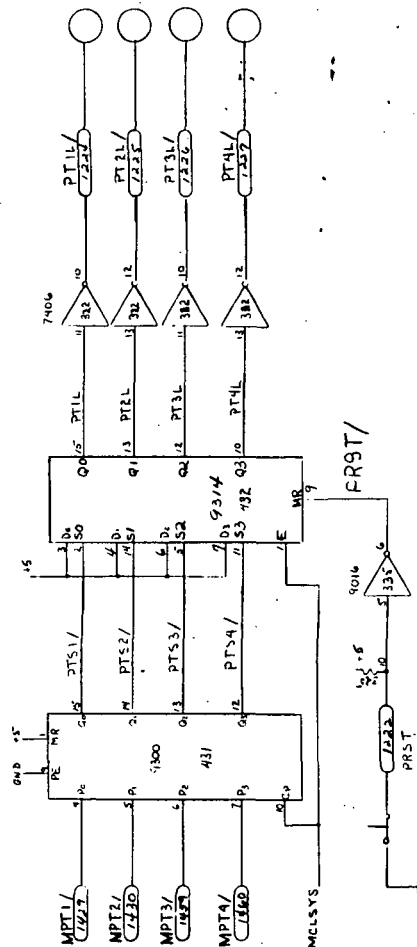
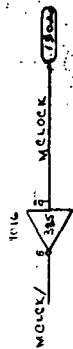
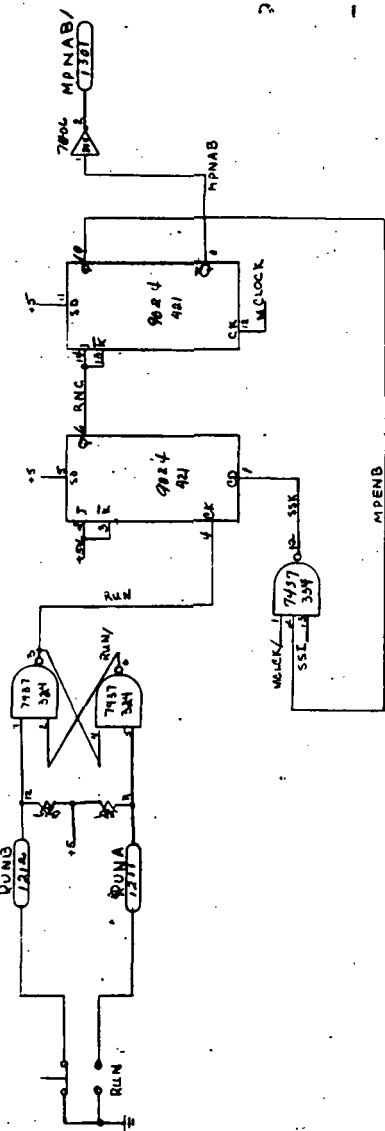






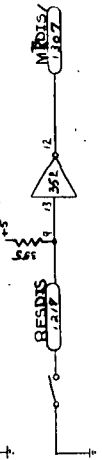
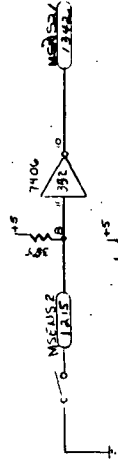
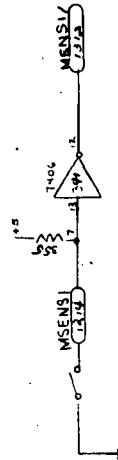
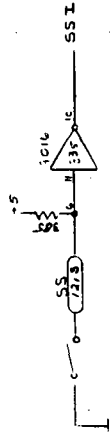
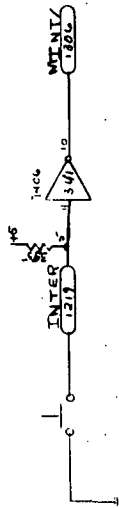
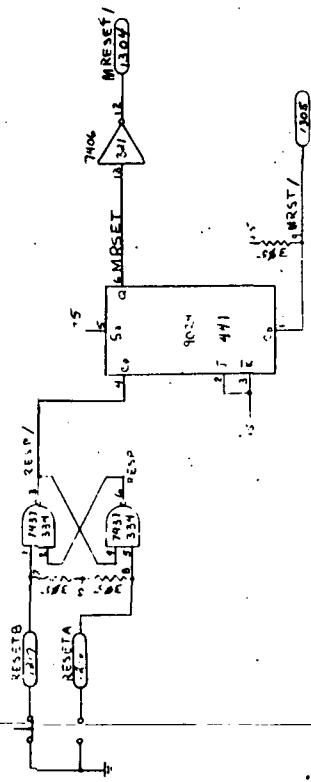




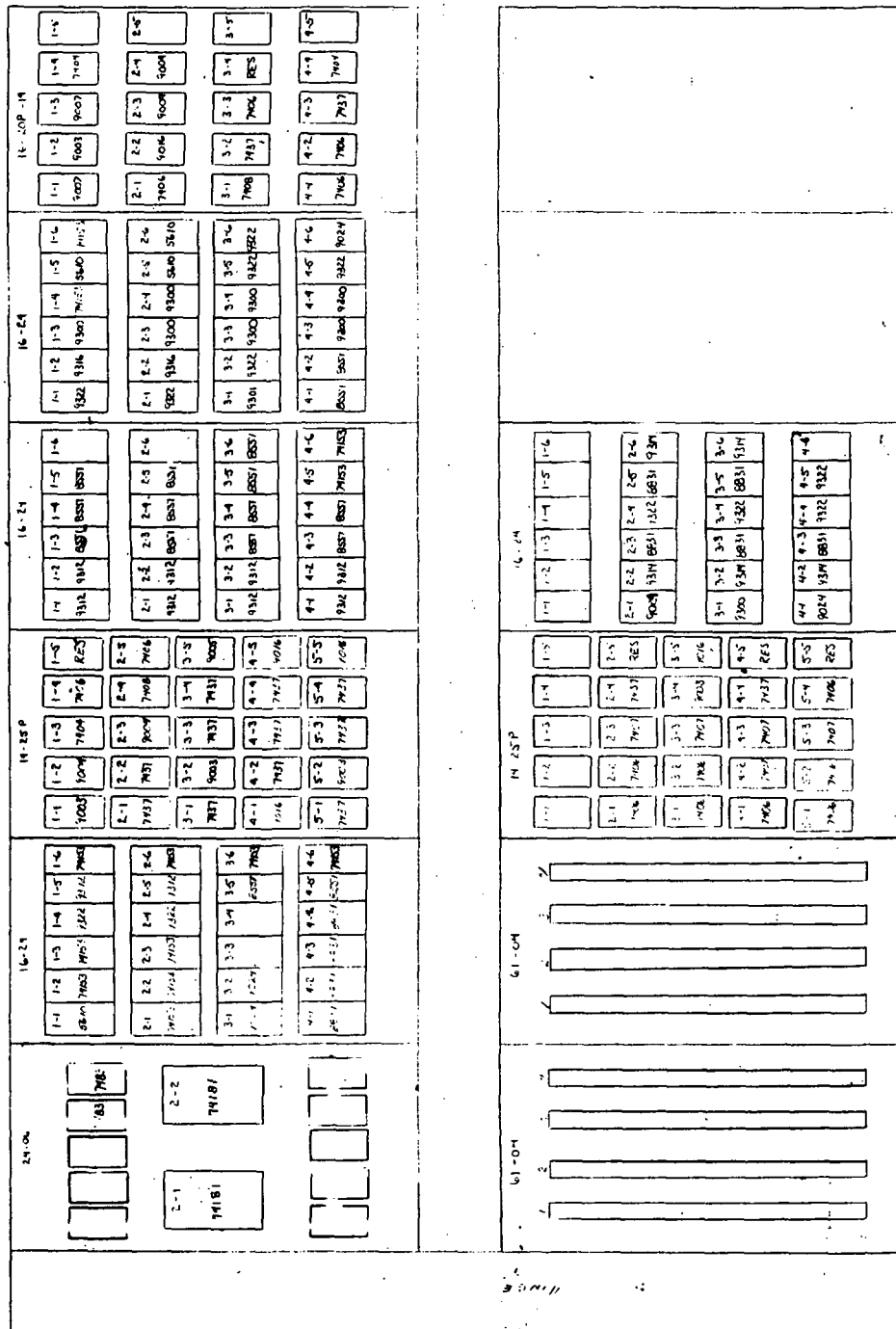


REV-F

Maintenance Panel Electronics



MAINT. PANEL CASE	
DATE	REV
BY	CHK
APP	APP
DATE	REV
BY	CHK
APP	APP



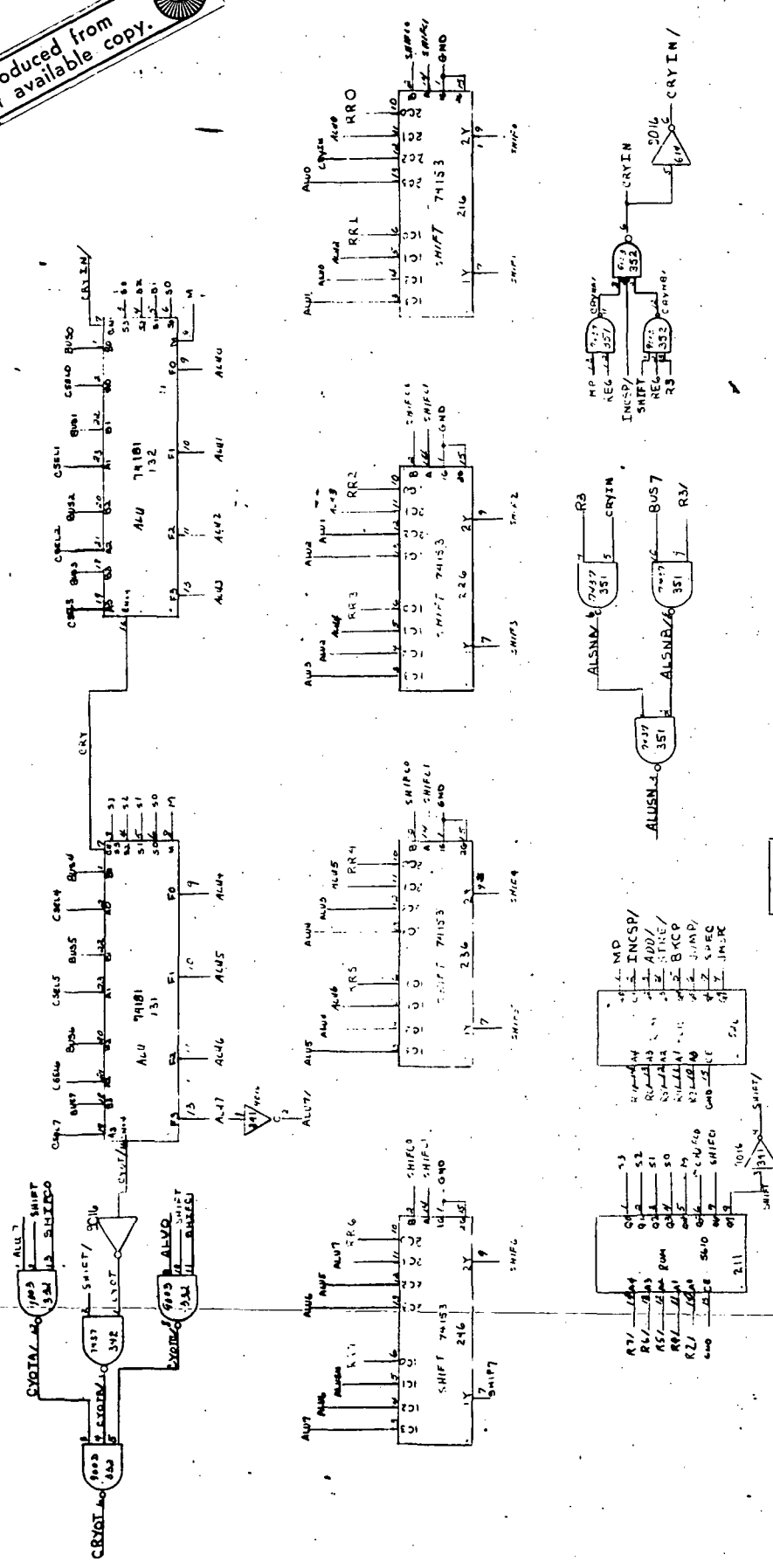
REV. =

Processor and Maintenance Electronics

I.C. Layout - Pin Side

NCR	
NATIONAL SEMICONDUCTOR CORPORATION	
SHEET 1 OF 1	
SCALE:	
CODE:	

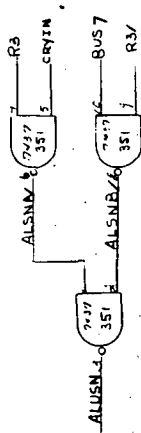
Reproduced from
best available copy.



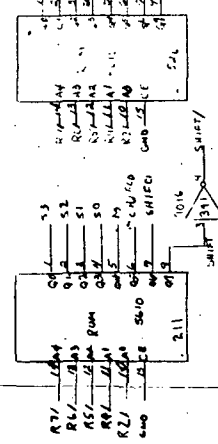
- BUS7 - 74105
- BUS6 - 74105
- BUS5 - 74105
- BUS4 - 74105
- BUS3 - 74105
- BUS2 - 74105
- BUS1 - 74105
- BUS0 - 74105

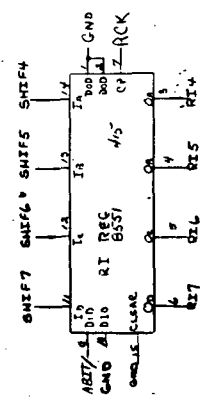
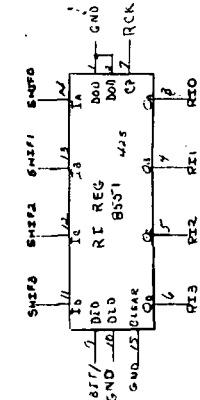
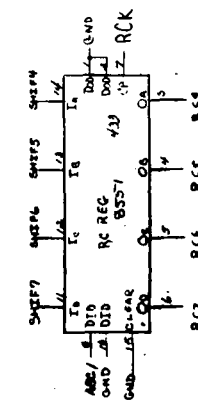
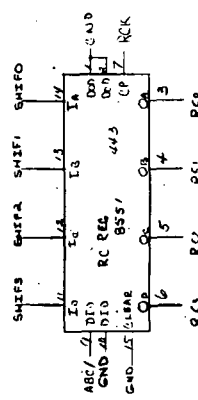
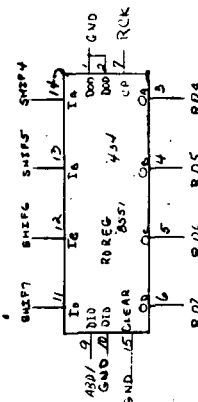
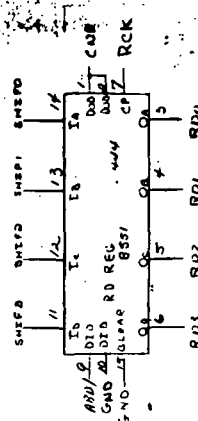
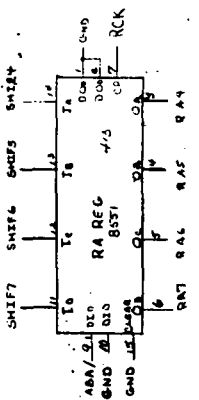
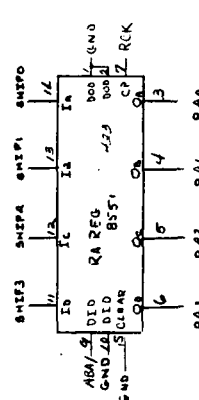
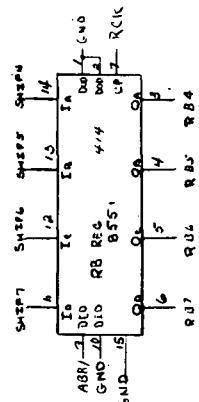
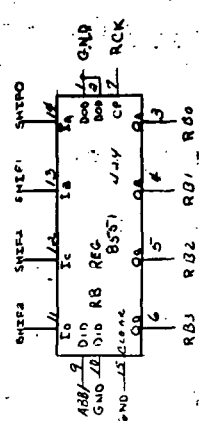
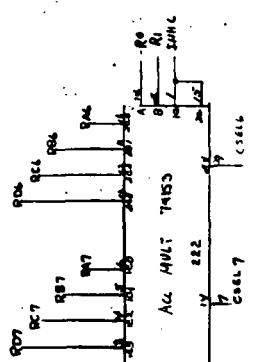
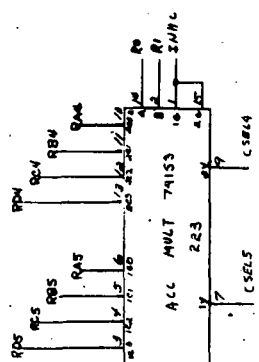
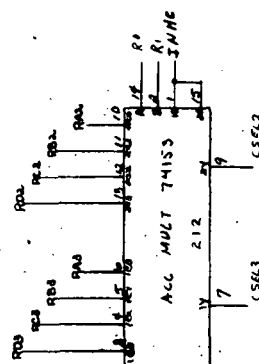
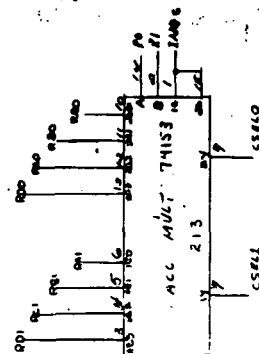


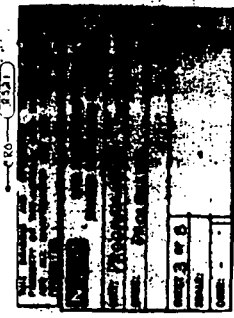
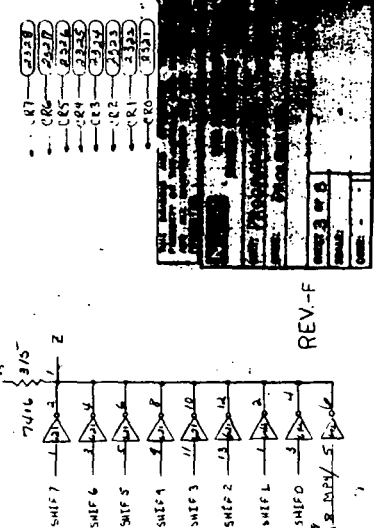
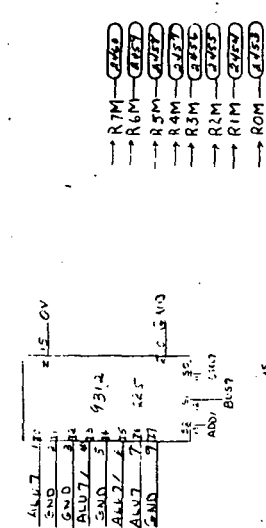
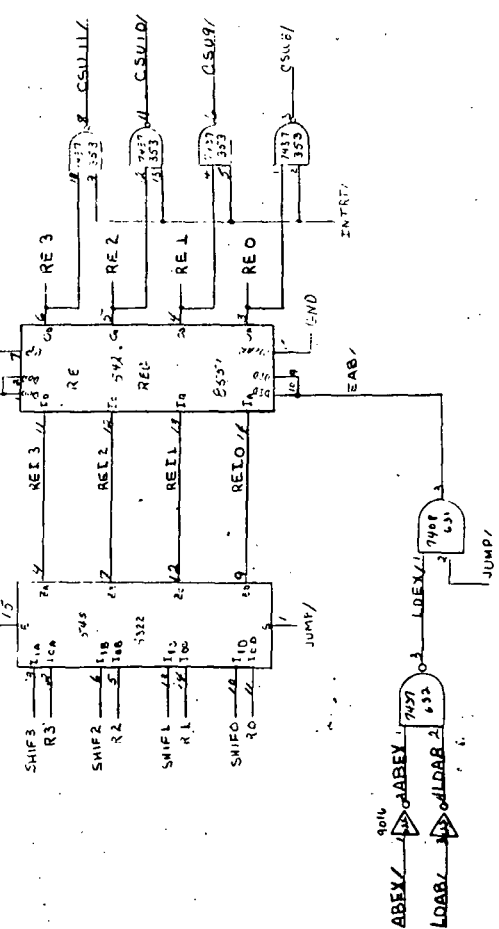
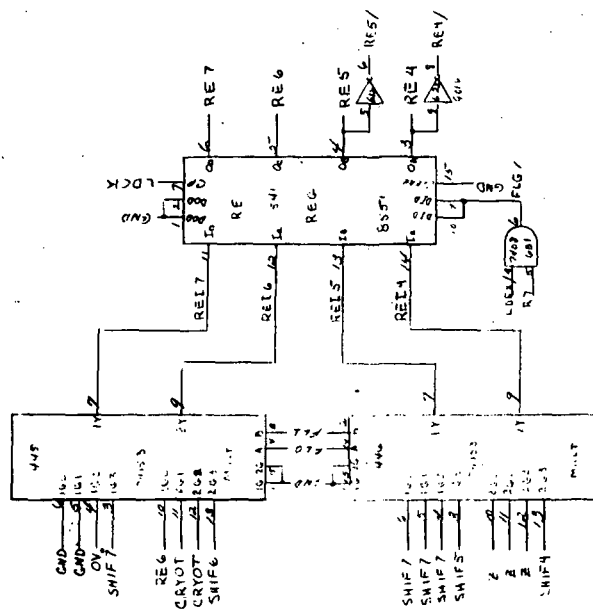
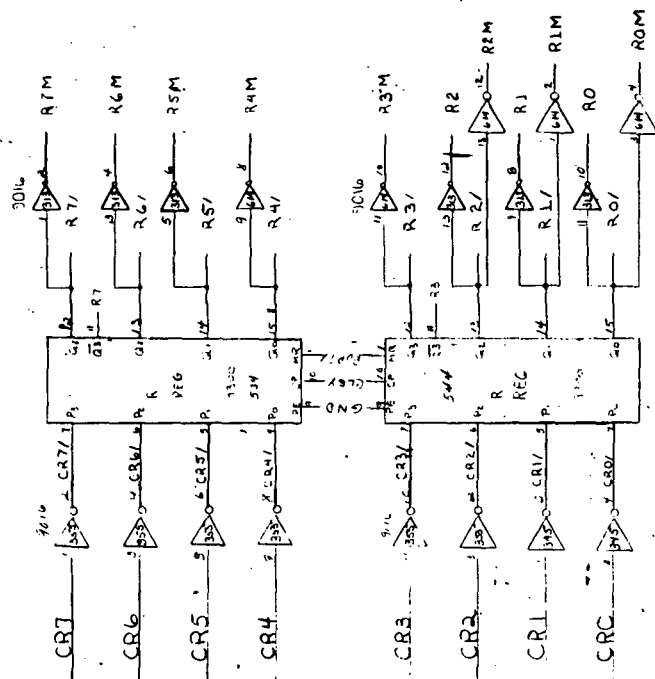
REV - F

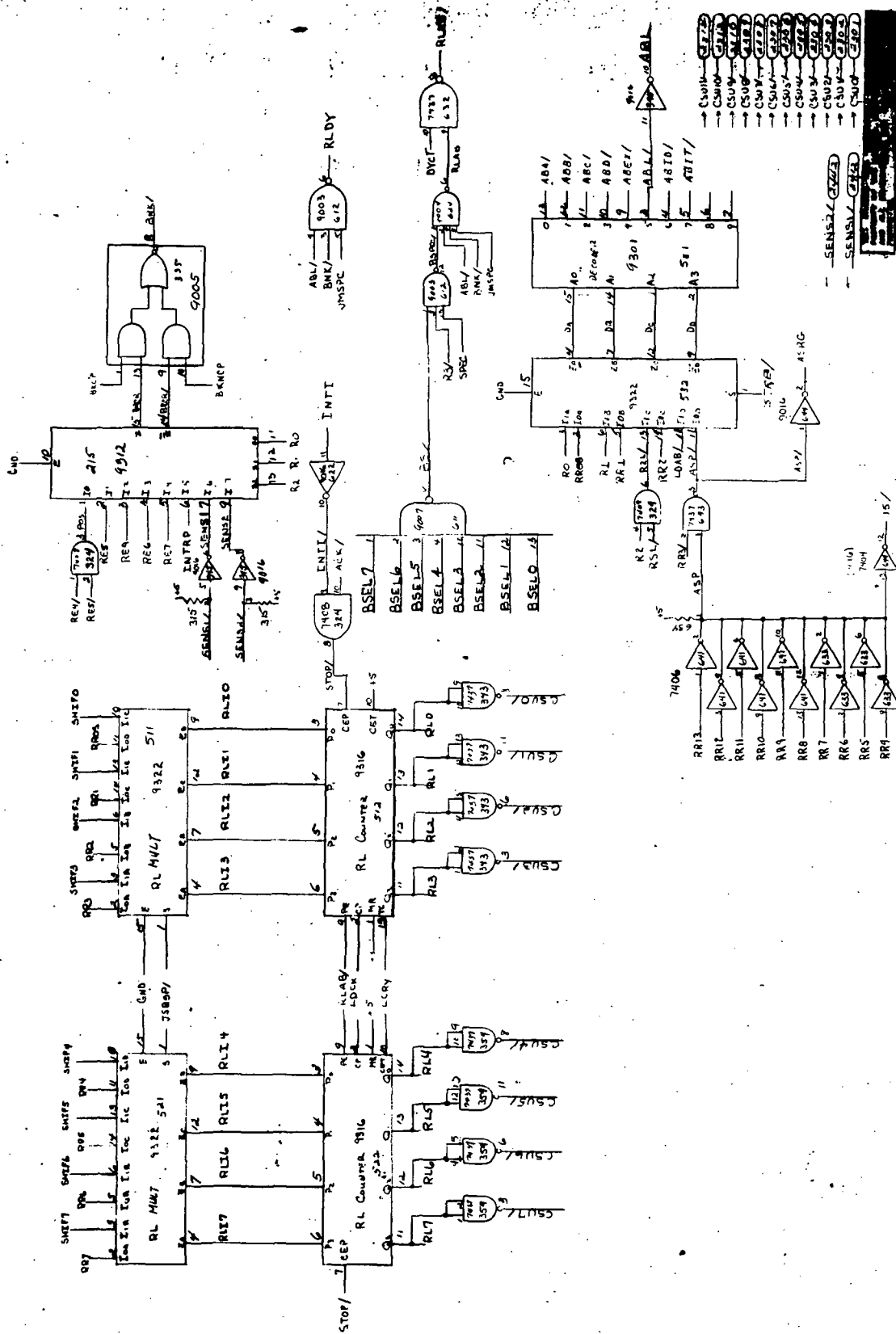


- BUS7 - 74105
- BUS6 - 74105
- BUS5 - 74105
- BUS4 - 74105
- BUS3 - 74105
- BUS2 - 74105
- BUS1 - 74105
- BUS0 - 74105

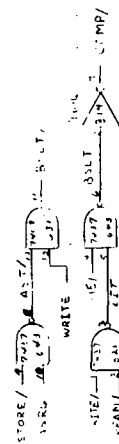
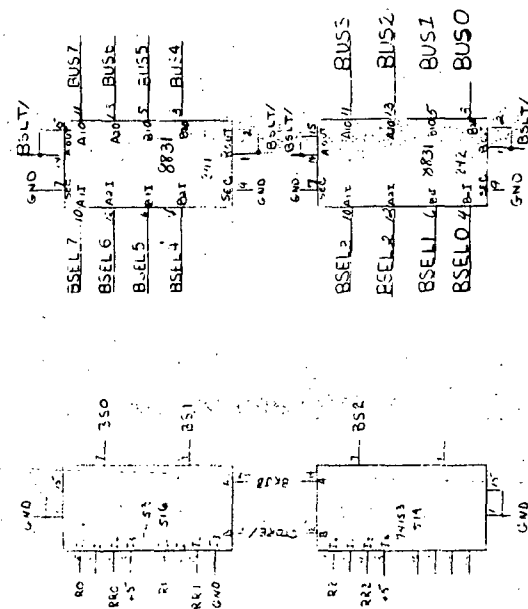








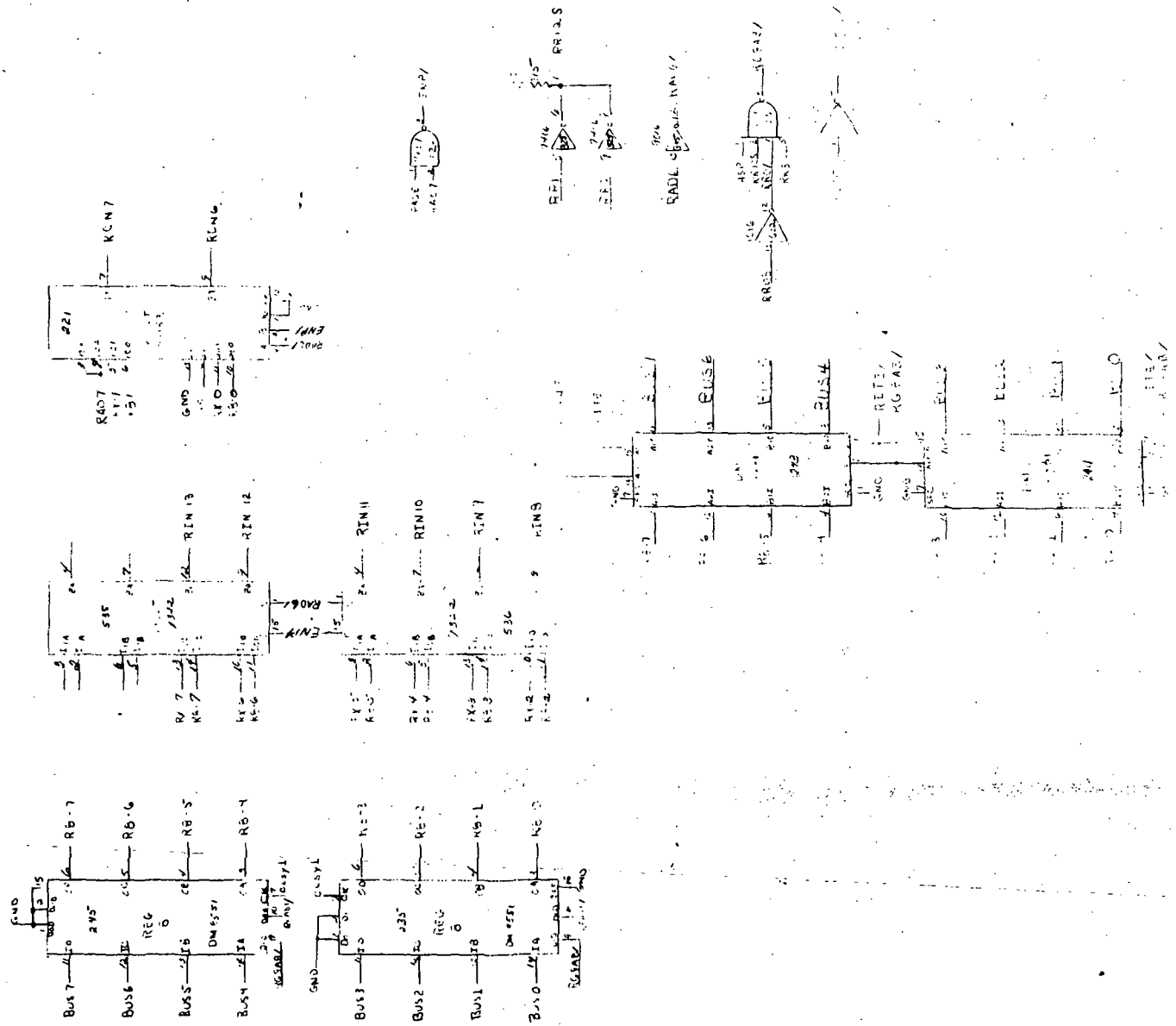
REV-F

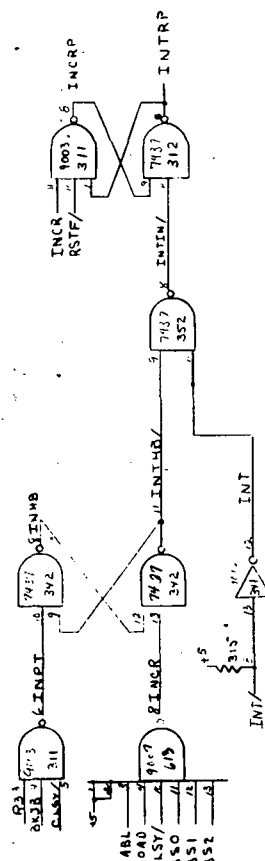
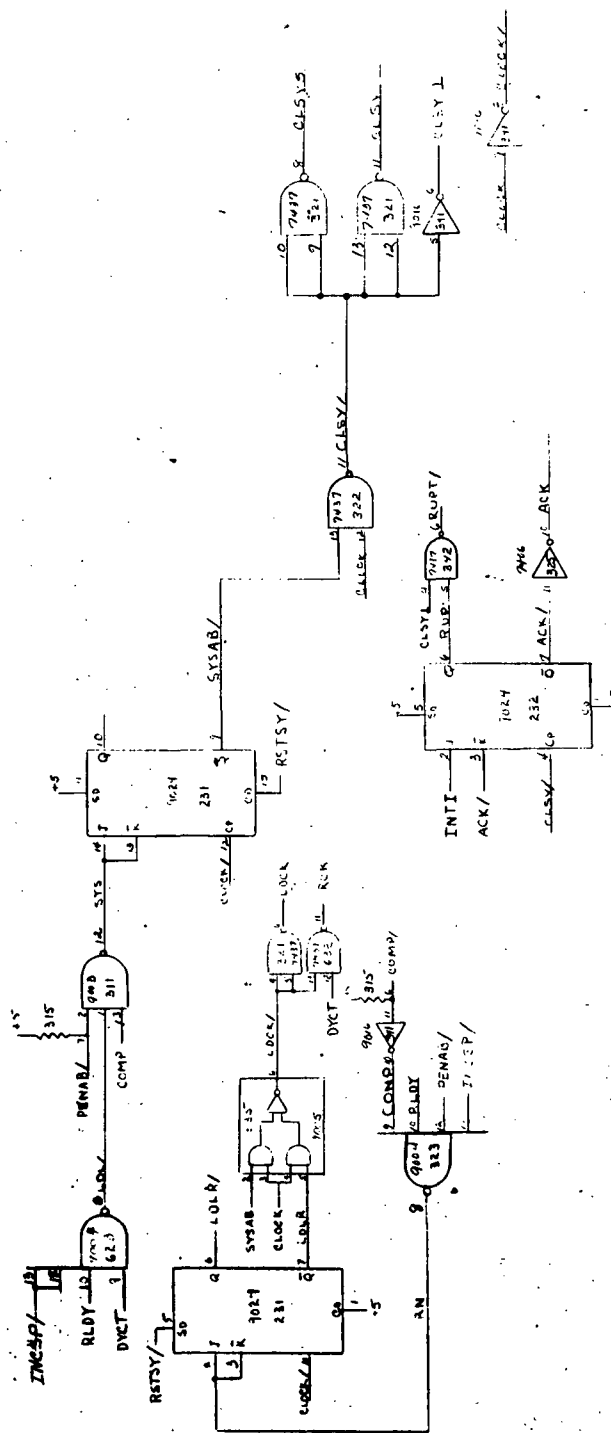
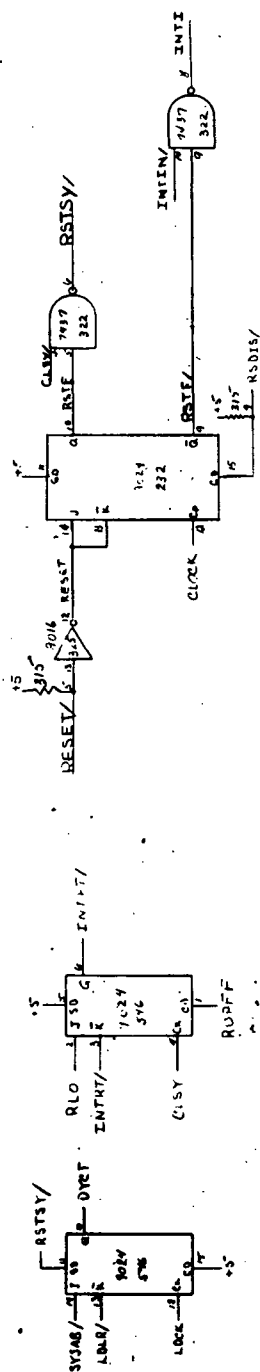
REV-11

Reproduced from
best available copy.

THIS DRAWING IS THE PROPERTY OF THE NATIONAL CANADIAN PATENT AND ALL INFORMATION CONTAINED HEREIN IS UNCLASSIFIED	
NCR DEN PROCESSING SYSTEM	
UNIT: PROGRAMABLE CONTROLLER	NAME: PROCESSOR
SHEET 7 OF 8	SCALE:
CODE:	REV:

REV - F



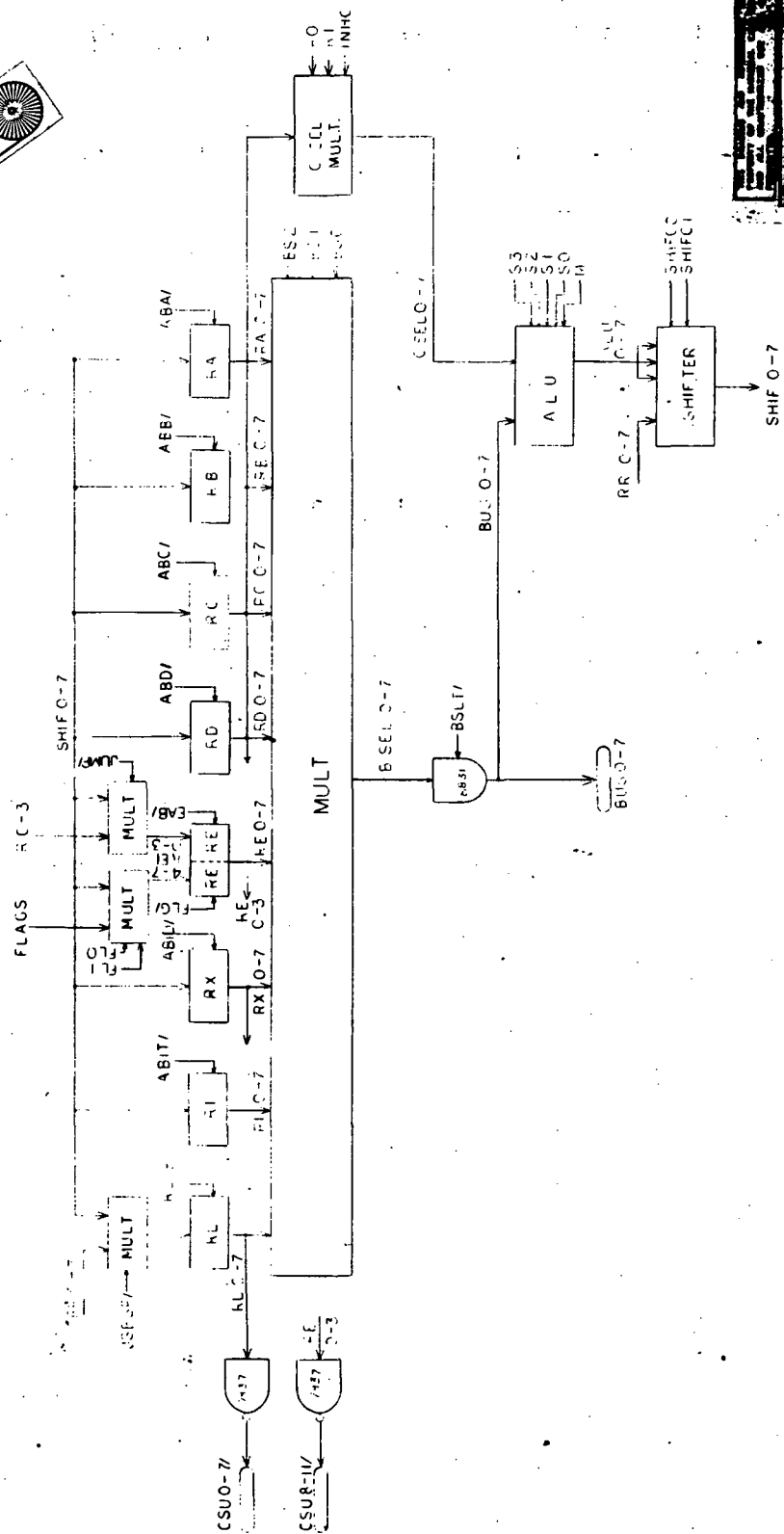


→ RSTSY/ — 322
 → ACK — 322
 → RSTSY/ — 322
 → RSTSY/ — 322
 → RSTSY/ — 322
 → RSTSY/ — 322

THIS DRAWING IS THE PROPERTY OF THE NATIONAL CANADIAN PATENT OFFICE. IT IS LOANED TO YOU FOR INFORMATION ONLY. IT IS NOT TO BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM.	
NCR NATIONAL CANADIAN PATENT OFFICE	
UNIT: PROGRAMMABLE CONTROLLER	NAME: PROCESSOR
SHEET: 8 OF 8	SCALE:
CODE:	REV:

261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

Reproduced from
best available copy.



REV	DATE	BY	CHKD	APPD	REMARKS
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					
40					
41					
42					
43					
44					
45					
46					
47					
48					
49					
50					
51					
52					
53					
54					
55					
56					
57					
58					
59					
60					
61					
62					
63					
64					
65					
66					
67					
68					
69					
70					
71					
72					
73					
74					
75					
76					
77					
78					
79					
80					
81					
82					
83					
84					
85					
86					
87					
88					
89					
90					
91					
92					
93					
94					
95					
96					
97					
98					
99					
100					

REV F