BDX-613-1566
Distribution Category UC-38

LABORATORY AUTOMATIC
DATA ACQUISITION SYSTEM

Published September 1977

Project Leader:
L. E. Schantz
Department 842

Project Team:
P. L. Blessner
R. D. Hensley

EP 04122656
Final Report

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Communications Services

**Kansas City
Division**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LABORATORY AUTOMATIC DATA ACQUISITION SYSTEM

BDX-613-1566, UNCLASSIFIED Final Report, Published September 1977

Prepared by L. E. Schantz, D/842, under EP 04122656

A laboratory automatic data acquisition system (ADAS) has been established in the electronic products laboratory. The system consists of three basic blocks: the software control calculator, which controls the entire system, receives data from the system, and acts upon it in some intelligent manner; a variety of laboratory measuring equipment which gathers data and inputs it in BCD format; and the coupler/controller-and-patch-panel combination which makes it possible for BDC information from the gathering apparatus to be transmitted to the calculator. Approximately 50 software programs have been written for the system, including one which permits the ADAS system to scan resistors, measure their values, record the values, and permit the percent change in resistor values to be plotted. A program has also been written for data on temperature coefficients of resistance on thin film resistors.

WPC-mlb

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

The Bendix Corporation
Kansas City Division
P.O. Box 1159
Kansas City, Missouri 64141

A prime contractor with the United States
Energy Research and Development
Administration under Contract
Number EY-76-C-04-0613 USERDA

CONTENTS

Section	Page
SUMMARY.	9
DISCUSSION	11
SCOPE AND PURPOSE.	11
PRIOR WORK	11
ACTIVITY	11
<u>System Overview.</u>	11
<u>System Description</u>	12
<u>Coupler/Controller</u>	17
<u>Patch Panels</u>	23
<u>BCD Input Card</u>	23
<u>BCD Output Card.</u>	30
<u>Clock/Timer/Pacer Card</u>	31
<u>Calculator Interface Card.</u>	32
<u>Pinboard Controller.</u>	36
<u>Teleprinter Card</u>	38
<u>Multiprogrammer and Multiprogrammer Extender</u>	40
<u>Sample Application: Tantalum Nitride Resistor</u> <u>Life Test.</u>	46
ACCOMPLISHMENTS.	49
FUTURE WORK.	50
APPENDICES	
A. DMM PATCHBOARD PROGRAMMING HP3409A BCD OUTPUT TO HP2570A SLOT 4 BCD INPUT CARD	57
B. PATCH PANEL AND REAR CONNECTOR WIRE LIST	59

C.	FUNCTIONAL INPUTS TO REAR CONNECTORS OF PATCH PANEL ASSEMBLY.	63
D.	DMM REMOTE CONTROL PATCHBOARD PROGRAMMING HP3490 DMM REMOTE CONTROL FROM BCD (NEGATIVE TRUE LOGIC) OUTPUT SLOT 2 (FORMAT "@2I1FAR1@4E").	65
E.	RELAY, DRVR, CONVERT, CODE AND CONVERT, VOLT, AND CLOCK READ SUBROUTINES.	67
F.	RESISTOR LIFE STABILITY TEST SUBROUTINE	79
	DISTRIBUTION.	83

ILLUSTRATIONS

Figure		Page
1	Automatic Laboratory Data Acquisition System Diagram.	13
2	ADAS Coupler/Controller and Patch Panels (Polaroid).	14
3	Input/Output and Control Devices (Polaroids).	15
4	Data Measurement and Test Apparatus (Polaroids)	18
5	Coupler/Controller System Design.	19
6	Coupler/Controller Internal I/O Hardware.	20
7	Typical System Operation Diagram for HP2570A Coupler/Controller.	22
8	Digital (Left) Patch Panel.	24
9	Analog (Right) Patch Panel.	25
10	Patch Panels (P-90292).	26
11	BCD Input Card (P-90296).	27
12	Computer Coded Date Calendar.	33
13	Calculator Command Formats.	35
14	Pinboard Controller Program Guide	37
15	Teleprinter Program Instructions and Controls.	39
16	Multiprogrammer and Multiprogrammer Extender (P-90303)	41
17	Interiors of Multiprogrammer and Multiprogrammer Extender (P-90294)	42
18	Multiprogrammer Word Formats.	44
19	Example of Relay Card in Multiprogrammer Extender (P-90305).	45

20	TaN Resistor Life Test Mainline Flow Chart. . .	51
21	TaN Resistor Life Test <i>Scan</i> Subroutine Flow Chart	52
22	TaN Resistor Life Test <i>INTL</i> Subroutine Flow Chart	53
23	TaN Resistor Life Test <i>List</i> Subroutine Flow Chart	54
24	TaN Resistor Life Test <i>Time</i> Subroutine Flow Chart	55

TABLES

Number		Page
1	Strobes for BCD Input Cards.	28
2	Bit Location on BCD Input Cable/Card/Patch Panel	29
3	Command Set for Calculator I/O Card	34
4	Multiprogrammer and Multiprogrammer Extender Output Slot Allocation.	43
5	Pin Data for 12 Individual Resistors on TaN Resistor Life Test Program.	47
6	Data Register Allocation for TaN Life Test. . .	48
7	Measurement Times and Relay Addresses for TaN Resistor Life Test.	49
8	Individual Data File Numbers for TaN Resistor Life Test.	50

SUMMARY

Until recently, data within the Hybrid Microcircuit Technology Laboratory had been acquired manually. As laboratory test setups became more complex and more numerous, the manual system began to over-burden the available manpower. Late in 1970 an automatic laboratory test and data acquisition system was proposed, but funding was not available at that time for what was conceived as primarily a hardware system. Funding became available in 1972, about the time calculator-based data acquisition and control systems came onto the market. These new calculator systems permitted the design of a *software controlled* system as opposed to the previously recommended hardware system. Work began immediately to select the system which would best fill the Bendix requirements for an automatic data acquisition system (ADAS).

The ADAS system was proposed to contain a Hewlett Packard 9820 calculator and Hewlett Packard 2570 coupler/controller as the basis of the system. To this system would be interfaced all of the laboratory measuring equipment in use at that time and all the equipment proposed for future acquisition. The coupler/controller, with its binary coded decimal (BCD) input and output (I/O) cards, permitted communication between data gathering equipment and the data analysis capabilities in the HP9820 calculator. However, instead of interfacing the data gathering equipment directly with the coupler/controller, it was proposed that the laboratory ADAS system use plug-in patch panels to connect the data gathering equipment and the coupler/controller.

It is the use of the patch panel which gives the system its excellent flexibility: a tremendous variety of laboratory apparatus having a BCD output can be coupled through the digital or analog patch panels to the coupler/controller and on to the calculator.

The ADAS system therefore consists of three basic blocks: the software control calculator, which controls the entire ADAS system, receives data from the ADAS system, and acts upon it in some intelligent manner; a variety of laboratory measuring equipment which gathers information and outputs it in BCD format; and the coupler/controller-and-patch-panel combination which makes it possible for BCD information from the gathering apparatus to be transmitted to the calculator.

The Hewlett Packard coupler/controller is the heart of the ADAS system. Internal to the coupler/controller are seven I/O slots which contain specific I/O functional cards. These cards control BCD input, BCD output, teleprinter input and output, and calculator interface input/output.

When the calculator is properly programmed to issue specific commands to the individual cards or slots in the coupler/controller, the specific functions on the I/O cards can be obtained.

Control of the ADAS system normally is delegated to the HP9820 calculator; however, control can be delegated to a pinboard controller located within the coupler/controller, or to a PDP11/05 minicomputer, or to a teleprinter. Only the HP9820 calculator and the PDP11/05 minicomputer can act on the data. The pinboard controller and the teleprinter card are limited to issuing commands to the coupler/controller; they lack the ability to analyse or interpret data.

The original justification for the ADAS system was based on work with life stability testing of thin film tantalum nitride resistors. Considerable time was involved in manually acquiring life stability data. It was proposed that the first use of the ADAS system be to automatically scan resistors, measure their values, record their values, and permit the percent change in resistor value to be plotted automatically. Such a program has been written, not only for life stability data on thin film resistors, but also for data on temperature coefficients of resistance.

Since the hardware portion of the ADAS system has been completed, approximately 50 software programs have been written for various test operations in the laboratory environment. All of these programs have been highly successful and have resulted in more efficient utilization of available manpower. It is anticipated that increase usage of the ADAS system will continue.

DISCUSSION

SCOPE AND PURPOSE

The laboratory automatic data acquisition system (ADAS) was proposed to simplify the time consuming and expensive procedure of manually gathering and reducing laboratory data for hybrid microelectronics technology development. A process development project was initiated to develop a small scale laboratory data acquisition system. Work on the project was done between July 1974 and December 1975.

This report is designed to acquaint potential users of the system with its capabilities, flexibilities, and limitations. It is not intended to be a complete operating manual for the ADAS system. An operating manual has been written and can be referred to by the user.

PRIOR WORK

Previous laboratory data acquisition has been manual; that is, the engineer has operated the test equipment, read the meter, recorded the measurements, and later entered the data into a computer or calculator for reduction and analysis. With single-shot tester operations, this procedure was adequate; however, as test setups became more complicated and as the quantity of parts under test increased, the shortcomings of a manual system became obvious. In 1970, it was proposed that an automatic laboratory test and data acquisition system be established for general laboratory usage, but no capital funding was available at that time. In 1972, funding became available about the time calculator-based data acquisition and control systems came onto the market. These new calculator systems permitted a software controlled system to be established rather than the previously proposed hardware controlled system.

ACTIVITY

System Overview

The automatic laboratory data acquisition system (ADAS) was designed to perform five tasks:

- Data acquisition,
- Data recording,
- Data reduction,

- Software controlled open set-up testing, and
- Hybrid microcircuit testing.

The ADAS can accomplish any one or a combination of these tasks. Since the system is designed to interface a variety of laboratory apparatus (Figure 1), the interface hardware must be extremely flexible. The laboratory apparatus is extremely varied in its output hardware format, although most laboratory apparatus has a binary coded decimal (BCD) output. If this BCD output is available, the apparatus can be interfaced with the ADAS.

Shown at the bottom of Figure 1 are the laboratory test apparatus which must be interfaced to the calculator or computer systems which in turn receive and reduce the data. The interface is accomplished through the items shown in the center of Figure 1: the Hewlett-Packard 2570A coupler/controller and the digital and analog patch panels. Providing a secondary interface and stimulus source between the measuring apparatus and the calculators are the Hewlett Packard multiprogrammer and multiprogrammer extender (HP6936A and HP6937A). The multiprogrammer can be thought of as an extension of the coupler/controller with only output capabilities.

Figure 2 shows the ADAS coupler/controller and the patch panels in a three bay assembly. In the center bay at the bottom are the digital (left) and analog (right) patch panels. Immediately above the patch panels (reading up) are the HP2570 coupler/controller, an HP3490 digital multimeter, the HP6936A multiprogrammer, and the HP6937A multiprogrammer extender. In the left and right bays are several of the digital and analog laboratory test apparatus.

System Description

The basic purpose of a laboratory data acquisition system is to interface "data gathering" apparatus with "data reduction" apparatus. This is shown schematically in Figure 1.

The coupler/controller and the digital and analog patch panels shown in the center of Figure 1 are the heart of the ADAS system: the coupler/controller and the patch panels allow the data measuring and data recording/reduction apparatus to "talk" to each other. The patch panels provide the flexibility for interfacing one or several of the test apparatus shown at the bottom of Figure 1 with the I/O units shown at the top of the figure. These I/O devices are operator-system interfaces and should not be confused with the coupler/controller I/O cards which allow different parts of the ADAS to communicate with each other. Therefore, it is the combination of the coupler/controller and the patch panels which provides the tremendous flexibility in this system.

Figure 1. Automatic Laboratory Data Acquisition System Diagram

Figure 2. ADAS Coupler/Controller and Patch Panels

The following four I/O devices are shown schematically in Figure 1 and pictorially in Figure 3:

- The PDP11/05 minicomputer,
- The HP9820A calculator,
- A standard teletype (or teleprinter), and
- A pinboard control panel (part of the coupler/controller).

All four devices are capable of controlling the tester through either software or hardware commands. However, only two of the devices are capable of reducing or acting upon data: the PDP11/05 minicomputer and the HP9820 calculator. The teletype is capable of issuing commands to the system, obtaining data, and then printing the data on paper. The pinboard controller may issue commands to the system in a fixed sequence and repeat that sequence; it is not capable of receiving data, outputting data by itself, making decisions, or reducing data.

Since it is the ultimate goal of the engineer to reduce the data obtained, attention shall be directed to the HP9820 calculator. It is capable of issuing commands to the system, receiving information from the system, and reducing or acting upon the data in some intelligent manner.

Figure 3. Input/Output and Control Devices

Figure 3 Continued. Input/Output and Control Devices

Most of the data gathering apparatus already interfaced with the ADAS is shown at the bottom of Figure 1 and in Figures 2 and 4. Measuring equipment shown in Figure 2 includes, in the left bay beginning at the top, a function generator, a digital counter, a sweep frequency generator, and a programmable pulse generator, and in the right bay, a Tektronics 576 oscilloscope and a Type 230 digital unit. Located below the oscilloscope is a stimulus panel.

Several of many data measuring apparatus are shown in Figure 4. The Caviderm is used to measure the through-hole resistance of vias in thin film substrates. The Stratham oven is a programmable temperature chamber which has been adapted through the connectors on the oven door to interface with several relays in the multi-programmer. The digital micrometer is used to measure linear distances on thin film substrates or any other part measuring less than 25 mm. The general HMC test fixture has been designed to receive one hybrid microcircuit package which then can be interfaced to the patch panel on the test fixture itself and to the patch panel of the ADAS system. The commonality between individual measurement apparatus is the BCD output. It is because of the BCD output that the test apparatus can be interfaced to the coupler/controller.

With the general HMC tester and several other test apparatus it is possible to provide stimulus to the hybrid circuit under test or to any other device under test through the function generator HP3300A, the Tektronic pulse generator, and the various power supplies built into the ADAS system.

Coupler/Controller

The coupler/controller will accept eight interface cards numbered I/O-0 through I/O-7 and one pinboard controller card. These eight slots in the coupler/controller are shown schematically in Figure 5 and pictorially in Figure 6.

In Slot 0 is the teletype interface card. In Slots 1 and 4 are BCD input cards. In Slots 2 and 3 are BCD output cards. Slot 5 has a time share interface card which interfaces the ADAS system with the CDC6600 computer. Slot 6 has a clock/timer/controller card and Slot 7 has the calculator interface card. It can be seen from Figure 5 that both the BCD input cards and one of the BCD output cards go directly to the patch panels. The remaining BCD output card is hardwired directly to the multiprogrammer. It is through the patch panels that all the data measuring apparatus interface the coupler/controller.

To transfer a reading from the digital multimeter (DMM) to the calculator, for example, the DMM output data (BCD format) would

Figure 4. Data Measurement and Test Apparatus

Figure 5. Coupler/Controller System Design

have to first go through the digital patch panel into one of the BCD input cards and then to the calculator. To set the function generator to perform a particular function, on the other hand, information would be programmed into the software of the calculator, the calculator would output commands to the BCD output card which would in turn strobe this information through the digital patch panel to the function generator--which would then perform the requested function.

For the calculator to operate in a closed loop--to set and verify the frequency of the function generator, for example--the following software program could be set up.

I/O SLOT ALLOCATION FOR HP2570A COUPLER/CONTROLLER

SLOT NUMBER	ALLOCATION
-	CONTROL BOARD (02570-60002)
0	TELETYPE (12801-66501)
1	BCD INPUT (5100-6703)
2	BCD OUTPUT (12798-60011)
3	BCD OUTPUT (12798-60011)
4	BCD INPUT (12797-60001)
5	TIME SHARE (12809-60001)
6	CLOCK CONTROL (12811-60002)
7	CALCULATOR INTERFACE (12822-60006)

Figure 6. Coupler/Controller Internal I/O Hardware

- Program the calculator to give a particular frequency and a particular voltage amplitude for a sine-wave.
- Output this command through the coupler/controller to the function generator.
- Check the function generator by looking at the output frequency of the generator on the counter and the voltage amplitude on the digital multimeter.
- Feed this information back through the patch panel and BCD input cards of the coupler/controller to the calculator and compare it to the desired values. If the measured values from the digital multimeter and the counter are the same as those desired, proceed with the next part of the program or stop. If the measured values and the desired values do not agree, jump to a subroutine which looks at the error between the two signals and outputs another command to the function generator to correct either in a positive or in a negative manner.

It would therefore be possible to output a command and then check to see if the command was obeyed or if it was interpreted properly.

The interface between the calculator and the coupler/controller is Serial ASCII. The ASCII command is taken into the coupler/controller by the calculator interface card located in Slot 7. The ASCII command is then transmitted onto the back plane (an 8 bit, parallel bus) of the coupler/controller. In the format shown in Figure 7, the @ 3E command received from the calculator into the calculator interface card is strobed out onto the back plane in that sequence. The @ symbol alerts the coupler/controller interface cards that a command is about to follow. The 3 designates the slot to which the command is directed and the following letter is a specific command. Therefore, when the @ 3 command is given, Slot 3 is alerted to listen for its command. All other card slots ignore the command that follows.

In the example given in Figure 7 a BCD input card is in Slot 3. Attached to this BCD input card, through a patch panel, is a digital voltmeter. In Slot 4 is a BCD output card which is in turn connected to a digital recorder such as the HP5050B.

The sequence of commands required to perform a "measurement and record" of the data is shown in Figure 7.

- The @ 3E means to encode the data source connected to the BCD input card in Slot 3. When this command is incurred the digital voltmeter is strobed to take a reading. This reading is transferred to the BCD input card and held there.

Figure 7. Typical System Operation Diagram for HP2570A Coupler/Controller

- The @ 4I command tells the BCD output card in Slot 4 to input the next data that is transferred onto the coupler/controller back plane and then to output that data to an external recorder.
- The @ 30 tells the BCD input card in Slot 3 to output the data which it had received previously from the digital voltmeter onto the back plane of the coupler/controller. As this data is strobed serially onto the back plane from Slot 3, Slot 4 is listening to the data and latches it into its registers, one data word (eight data bits) at a time. The BCD output card in Slot 4 then outputs the data to the printer and the program sequence is complete.

All other interfaces through the coupler/controller between the calculator and any other piece of measurement hardware are performed in a similar manner.

The example in Figure 7 does not actually show the data going through a digital patch panel. In fact, the BCD output and BCD input cards are hardwired to the printer and the digital voltmeter respectively. This is undesirable in generalized laboratory ADAS systems. In generalized systems, it is possible to interface a variety of data measuring apparatus by making a hardwire cable to interface between the BCD input cards and each of the data gathering apparatus. However, this would be unnecessarily bulky and would require a great many cables. What can be done (Figure 5) is to break the hardwire cable between the BCD input cards and the data measuring apparatus. This break occurs at the digital and analog patch panels. To interface the digital voltmeter and the BCD input cards, for example, one need only plug-in the appropriate patch panel connections. Appendix A discusses specific DMM-BCD patch board connections.

Patch Panels

Figures 8, 9, and 10 show the digital and analog patch panels. As can be seen in Figure 8, BCD input from Slot 4 is located in Rows A and B, Columns 1 through 20, and Row P, Columns 13 through 20. The BCD input card located in Slot 1 is terminated in Rows J and K, Columns 1 through 20, and Row P, Columns 1 through 8. The BCD output card located in Slot 2 terminates on the patch panel in Rows L and M, Columns 1 through 20.

The digital measurement apparatus also terminates on the digital patch panel. Therefore, to interface the digital multimeter shown in Columns C and D with the BCD input card located in Slot 4 the connections are jumpered from Row C and D to Rows A and B. Specific patch panel connections to accomplish these interfaces are shown in Appendix A.

BCD Input Card

The coupler/controller shown in Figure 6 presently has two BCD input cards. These cards are located in Slots 1 and 4. All BCD data which is to be transferred from a measuring instrument to the calculator must be transmitted via one of the BCD input cards. A cable is connected to the end of the cards in all the slots (Figure 6); in the case of the BCD input cards these cables terminate on the digital patch panel as shown in Figure 8.

Each BCD input card is a 40 bit interface. Since it is a binary coded decimal, each decimal character requires 4 bits, permitting 10 decimal digits to be represented on this card. In Figure 11,

Figure 8. Digital (Left) Patch Panel

the first row of integrated circuits are latches which hold the binary coded information as it is received by the card. There are 10 latches, each one "holding" a decimal digit. Transmission to the BCD input card is in parallel; that is, all 40 bits or 10 decimal characters are taken onto the card simultaneously. Since the back plane, or the communication between the BCD input card and any other device such as the calculator interface card, is a serial transmission, the decimal characters must be strobed onto the back plane one at a time by internal strobes. These strobes are programmed by the hardware pin connectors shown in the lower right corner of Figure 11. There are 13 strobes, each of which can be hardwired to a particular decimal location on the BCD input card. For example, to strobe out Character 1, one must connect Strobe 1 to Location 1, and, to do it in sequence, Strobe 1 must be hooked to Character 1, Strobe 2 to Character 2,

Figure 9. Analog (Right) Patch Panel

Strobe 3 to Character 3, and so on. The manner in which the information is strobed out is a function of two things:

- The way the data is to be taken into the collecting device, and
- The way the data is received and how it is hardwired into particular locations on the BCD input card.

Assume that the most significant decimal character (four bits of BCD data) is patched into Location 1, the next most significant character into Location 2, and so on down to Location 10. Further assume that the information is to be transmitted to the calculator, which requires that the information go in the order of most significant character first. Therefore, the strobes on the BCD input card would be as shown in Table 1 (for Slot 1, I/O-1): Strobe 1 goes to the most significant character; 2 would go to 2;

Figure 10. Patch Panels

3 would go to 3, 4 would go to 4, on down to where 10 would go to 10, which is the least significant character. The calculator must then be told that the word has ended. Strobe 11 would be to *end-of-word* (EOW), and then the EOW would go to *comma*, which is the word delimiter for the calculator.

This example is fine for fixed point or integer data; however, most data has a decimal point located somewhere within the string of digits. Where the range or the decimal point may be variable on the data being transmitted to the calculator, the strobing shown in the Slot 4 (I/O-4) portion of Table 1 would be used. The only difference is the strobing in nine decimal data characters. The tenth strobe puts in an exponential (represented by E), the eleventh character strobes in a negative sign, and the twelfth character strobes in the exponent or the range which had been strobed in BCD Location 10. This is an exponential input format: the most significant character first, down through the least significant character, followed by the exponential sign, a negative sign, and the exponential value. Immediately following this is the EOW and *comma*. The negative sign moves the decimal point to the left a number of places equal to the range/exponent value stored in Location 10.

This is useful with such things as the digital multimeter which outputs six characters and a range position. Each character is

Figure 11. BCD Input Card

represented by four bits, as is the range position. The most significant character from the digital multimeter is placed into BCD Location 4. Since there are only six decimal characters, the first three BCD locations are not used and will be assumed to be 0. The range character from the digital multimeter is placed into Location 10.

The example below shows BCD encoded data and now it is interpreted by the calculator.

BCD Encode Data 145673E-3

Calculator Interpreted Data: 145.673

Unless the calculator is told what the units of the measurement are, it will have no way of knowing. Where the measuring instrument does not indicate units and the units are important, such information must be prestored in the calculator software.

Table 2 shows the bit location on the BCD input cable/card/patch panel. It can be seen that there are groups of four patch panel designations for each decimal number that correspond to Storage Locations 1 through 10.

Table 1. Strobes for BCD Input Cards

Strobe	Location/Character
Slot 1 (I/O-1)	
1	1 (Most Significant)
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10 (Least Significant)
11	EOW →,
12	Open
13	Open
Slot 4 (I/O-4)	
1	1 (Most Significant)
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9 (Least Significant)
10	E
11	-
12	10 (Exponent/Range)
13	EOW →,

Table 2 can be used to fabricate a cable going from the measuring device to the patch panel by doing the following:

- Determine the number of decimal characters to be output from the measuring device,
- Fabricate a cable carrying four bits of data for each decimal character to be transferred,

Table 2. Bit Location on BCD Input Cable/Card/Patch Panel

Patch Panel Location		Storage Location	Bit	BCD Input Card Connector	Rear Patch Panel Connector
I/O-4	I/O-1				
B20	K20	1	1	1	40
B19	K19		2	2	39
B18	K18		4	A	38
B17	K17	1	8	B	37
B16	K16	2	1	3	36
B15	K15		2	4	35
B14	K14		4	C	34
B13	K13	2	8	D	33
B12	K12	3	1	5	32
B11	K11		2	6	31
B10	K10		4	E	30
B9	K9	3	8	F	29
B8	K8	4	1	7	28
B7	K7		2	8	27
B6	K6		4	H	26
B5	K5	4	8	J	25
B4	K4	5	1	9	24
B3	K3		2	10	23
B2	K2		4	K	22
B1	K1	5	8	L	21
A20	J20	6	1	11	20
A19	J19		2	12	19
A18	J18		4	M	18
A17	J17	6	8	N	17
A16	J16	7	1	13	16
A15	J15		2	14	15
A14	J14		4	P	14
A13	J13	7	8	R	13
A12	J12	8	1	15	12
A11	J11		2	16	11
A10	J10		4	S	10
A9	J9	8	8	T	9
A8	J8	9	1	17	8
A7	J7		2	18	7
A6	J6		4	U	6
A5	J5	9	8	V	5
A4	J4	10	1	19	4
A3	J3		2	20	3
A2	J2		4	W	2
A1	J1	10	8	X	1

- Attach this cable to the back of the patch panel connector box (Appendix B),
- List the decimal characters from the most significant to the least significant and the BCD bit weight associated with each of the four lines,
- Determine which patch panel location is to be assigned to each of the BCD bits (Table 2 and Appendices B and C), and
- Make an interconnection table to connect the termination of the measuring device cable to the BCD input card (Appendix A).

BCD Output Card

As can be seen in Figure 5, the coupler/controller has two BCD output cards located in I/O-2 and I/O-3. It can also be seen that only one BCD output card goes to the digital patch panel; the other is hardwired to the multiprogrammer. However, both cards are controlled by the calculator in the same manner: the BCD output card converts 10 decimal digits into 10 BCD equivalents for a total of 40 bits divided into 10 groups of 4 bits each.

The *I* command is the only basic command used by the BCD output card. When the @ 2I command is given from the calculator the BCD output card in Slot 2 responds by inputting decimal information from the calculator to the BCD card and outputting the equivalent BCD code. If, for example, an @ 2I 3 command is given to Slot 2, the BCD equivalent of the decimal digit 3, 0011, will be output. The remaining 9 decimal characters (36-bits) will be assumed to be 0 and a corresponding 0 BCD code will be output for those digits. Like the BCD input card, the BCD output card is a 40 bit card. Unlike the BCD input card, however, the BCD output card sends out the BCD information as the decimal digits are interrupted; that is, as each decimal digit is received serially by the BCD output card, a corresponding binary coded decimal is output and activates each group of four lines one at a time. Under special conditions the BCD output card can be made to store all 10 decimal characters and then send out the BCD information all in parallel, although this normally is not the case.

The BCD output card has these basic uses:

- To control the function and range of the digital micrometer, counter, voltmeter, or similar apparatus; and
- To provide 0 through 40 binary bits to control some external piece of hardware.

An example of the control of the digital multimeter by the BCD output card is shown in Appendix D.

Clock/Timer/Pacer Card

The clock/timer/pacer card located in Slot 6 provides a system clock for the coupler/controller. The 24-hour clock keeps time in hours, minutes, seconds, and tenths of seconds. A program source can order the printing of a current time on the system recording device--teleprinter, paper tape, or calculator. A different command can set current time in a buffer for later recording.

The timer function of the clock allows the programmer to introduce time delays into the system program. For example, a timed interval may be used after a change of stimulus to allow the system to "settle out" before making a measurement. The timer function may also be used to pace the system if the timed interval used is longer than the programmed execution time. The timed interval is selectable in 0.1 s steps over the range of 0.1 to 99.9 s.

The clock also has a pacer function which may control the frequency of execution of the data acquisition (and control) operations. The system program may be started at paced period of 0.1, 1, 10, and 60 seconds, 10 minutes, 1 hour, 1 day, or any of the selected periods multiplied by a factor of two or three.

As with the BCD input card and the BCD output card, the clock/timer/pacer card responds to letter commands.

- An *H command* halts the system until a pre-selected clock rollover has occurred. The *H command* is associated with the pacing function of the clock/timer/pacer card, Rollover Selection Switch S1, and Jumper W6 with Multiple Selection Positions X1, X2, and X3.
- An *I command* loads the setting selected on Switches S2, S3, and S4 into the interval generator on the clock/timer/pacer card. The interval generator counters are allowed to time out from this initial setting; the interval generator is initialized, and again allowed to time out.
- The *L command* stores current time in buffer storage. The command may occur at the beginning, end, or anywhere in the program to flag an important event. An *L command* must be followed by an *M command* to make use of the stored time.
- The *M command* results in the serial transfer of ASCII time characters from the time buffers, where they were stored on an *L command*, to the data bus of the coupler/controller back plane. An *M command* not preceded by an *L command* performs no useful function.

- When an *N command* is acted upon, the system clock is set to time $\emptyset$, accumulated rollover periods are cleared out of the register, and the clock error indicator on the clock control panel will go out if it was lighted. A possible use of this command is to clear the clock within an *N command* and store time with an *L command* at the beginning of a process. Then, at the end of the process, an *M command* followed by an *O command* records the start and end times of the process.
- The *O command* stores current time of day in buffer storage and transfers the ASCII time characters to the data bus of the coupler/controller back plane. If the *O command* follows an *L command*, current time (not the time stored by the *L command*) will be recorded.

To set the current time into the clock card the following operation should be performed:

- Refer to the day-of-month and computer coded date calendar (Figure 12).
- Using the front panel control switches on the clock card and the day corresponding to the current day of the year (Figure 12), set each digit into the clock control card by setting the switch to the day and pressing the set switch. The run switch should be in the *stop* position.
- Set the run switch on the *run* position and continue.

Calculator Interface Card

The calculator interface card is located in Slot 7 of the coupler/controller. It permits serial transmission of data to and from the HP9820 calculator, to the coupler/controller, and on to other external devices. There are seven letter commands for the calculator (Table 3).

The general command format is to input an @ AC where the @ alerts the coupler/controller that a command is about to follow, the A designates the number of the I/O slot to be addressed, and the C contains the desired command. The A is always a decimal number between $\emptyset$ and 7 and the C is always a valid alpha code for that particular I/O card.

For example (Figure 13), the command *FMT "@ 6L @ $\emptyset$ I @ 70 @ 6M"*, Z; WRT 4 outputs clock time from the clock card to the teletype and prints it on the teletype. The @ 6L tells the clock card located in Slot 6 to log the time, @ $\emptyset$ I tells the teletype to print or input the next data word that it finds on the back plane

FEBRUARY (2nd Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			032	033	034	035	036	037	038	039	040	041	042	043	044	045	046
16	17	18	19	20	21	22	23	24	25	26	27	28	29	Leap year only. Add 1 to each 3-digit number starting March 1.			
047	048	049	050	051	052	053	054	055	056	057	058	059	060				
MARCH (3rd Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			060	061	062	063	064	065	066	067	068	069	070	071	072	073	074
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
075	076	077	078	079	080	081	082	083	084	085	086	087	088	089	090		
APRIL (4th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			091	092	093	094	095	096	097	098	099	100	101	102	103	104	105
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30			
106	107	108	109	110	111	112	113	114	115	116	117	118	119	120			
MAY (5th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151		
JUNE (6th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			152	153	154	155	156	157	158	159	160	161	162	163	164	165	166
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30			
167	168	169	170	171	172	173	174	175	176	177	178	179	180	181			
JULY (7th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			182	183	184	185	186	187	188	189	190	191	192	193	194	195	196
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212		
AUGUST (8th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			213	214	215	216	217	218	219	220	221	222	223	224	225	226	227
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243		
SEPTEMBER (9th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			244	245	246	247	248	249	250	251	252	253	254	255	256	257	258
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30			
259	260	261	262	263	264	265	266	267	268	269	270	271	272	273			
OCTOBER (10th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			274	275	276	277	278	279	280	281	282	283	284	285	286	287	288
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304		
NOVEMBER (11th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30			
320	321	322	323	324	325	326	327	328	329	330	331	332	333	334			
DECEMBER (12th Month)			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			335	336	337	338	339	340	341	342	343	344	345	346	347	348	349
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365		

Figure 12. Computer Coded Date Calendar

Table 3. Command Set for Calculator I/O Card

Command	Description
A	<i>Abort.</i> Provides a <i>system normalize</i> to all I/O cards in the system, resetting the system. If given when the calculator is not the system controller, control is transferred to the system.
C	<i>Control.</i> Takes control of the system and enables the calculator to be the source of all commands required by the system.
F	<i>Format.</i> Used the program devices interfaced to the coupler/controller by either a relay register card or a BCD output card. The <i>F command</i> outputs specially formatted data capable of having leading zeros. The specified field width must be one space greater than the number of digits to be output.
I	<i>Input.</i> Inputs data to the calculator. When given by the calculator, the <i>I command</i> must be followed by either an <i>O</i> or <i>M command</i> addressed to the I/O card interfacing the data source. When given by some other system controller, the <i>I command</i> may be followed by any output command addressed to the data source I/O card.
K	<i>Kontinuous.</i> When given by the calculator, the <i>K command</i> is used with an <i>F</i> , <i>O</i> , or <i>I command</i> to provide continuous output or input of data words. When given by some other system controller, the <i>K command</i> is used only with an <i>I command</i> . The <i>K command</i> prevents a data word delimiter from cancelling an <i>F</i> , <i>O</i> , or <i>I command</i> . The <i>K command</i> is cancelled by an <i>N command</i> .
N	<i>Negate.</i> Cancels and <i>F</i> , <i>I</i> , <i>K</i> , or <i>O command</i> previously given to the card.
O	<i>Output.</i> Outputs fixed or floating point data from the calculator.

SYSTEM COMMAND:

FMT"@6L@0I@70@6M",Z;WRT4+
FMT"@0C",Z;WRT4H
FMT"@4E",Z;WRT4H

(OUTPUTS CLOCK TIME TO TTY)
(TRANSFERS CONTROL TO TTY)
(CAUSES DMM TO TAKE A READING
[ENCODE])

DATA INPUT COMMAND:

FMT"@4E@7I@40",Z;WRT4;RED4,R(X)+

(DMM ENCODES AND TRANSFERS
DATA TO CALCULATOR REGISTER
R[X])

DATA OUTPUT COMMAND:

FMT"@0I@70",FIX10.4,"",Z;WRT4,R(X)+

(DATA FROM CALCULATOR REGISTER
R[X] IS PRINTED ON TTY)

MULTIPROGRAMMER COMMAND:

FMT"@3I@7F",FIX9.0,"",Z;WRT4,R(X)+

(DATA OR CONTROL WORD IS
TRANSFERRED FROM CALCULATOR
REGISTER R[X] TO THE MULTI-
PROGRAMMER)

Figure 13. Calculator Command Formats

of the coupler/controller, the @ 70 tells the coupler/controller that the next command will be coming from the calculator and should not be interpreted as data, the @ 6M tells the clock card located in Slot 6 to output the recorded time onto the back plane, and the Z; WRT 4 is a necessary syntax for the calculator to transfer the information to the coupler/controller.

In the system command FMT "@ 4E", Z; WRT 4, the @ 4E is addressing I/O Slot 4, a piece of measuring equipment via the BCD input card. The E command tells the input card to have the measuring equipment take a reading. In the case where the BCD input card is connected to the digital multimeter via the patch panel, the multimeter takes one sampling or one reading and transfers that information to the BCD input card where it is stored and held.

Since it does little good to have the information from the digital multimeter stored only on the BCD input card, this information must now be transferred to either the teletype for printing or to the calculator for data reduction. To do this the next set of commands or the data input commands are used: FMT "@ 4E @ 4I @ 40", Z; WRT 4; Red 4, R(X). The @ 4E tells the BCD input card

in Slot four to take a reading. The @ 7I tells the calculator via the calculator interface card in Slot 7 to input the next data word on the coupler/controller back plane. And @ 40 tells the BCD output card to take the data previously recorded and output it onto the back plane of the coupler/controller. RED 4, R(X) tells the calculator to take the data from the calculator interface card and store it into Register X where X is any number or calculator register.

When the calculator has determined some information or piece of data and wishes to output it to some other area via the coupler/controller, a *data output command* is used. A data output command uses the format statement like a data input command. However, instead of inputting information to the calculator, the command outputs information from the calculator.

Assuming that the data to be output is stored in Calculator Register R(X), it can be transferred by using the command @ 70. First, the @ 0I command tells the teletype located or attached to the teletype card in Slot 0 to print the next data word that comes along on the back plane. The @ 70 command tells the calculator interface card to receive and transfer the forthcoming data. The FIX 10.4 says that the information coming through the calculator has 10 characters in it and has a decimal located 4 places to the left. The WRT 4, R(X) tells the calculator to output the data located in R(X) to the coupler/controller. From there the coupler/controller transfers this data along the back plane to I/O Slot 0. From I/O Slot 0 the information is passed to teletype where it is printed on the teletype paper.

The last category of command is the multiprogrammer command which transmits a command or a data word from the calculator to the multiprogrammer.

Pinboard Controller

There is a pinboard program controller on the far left side of the coupler/controller shown in Figure 6. The actual pinboard control board is shown in Figure 3. The pinboard controller is a manually-programmed alternative program control device. In the upper portion of the board is a series of parallel "plug-ins" which hold diode pins. A suitable arrangement of diode pins represents an @ AC command.

There are 15 possible program steps on the pinboard (Figure 14). It is possible to allow the coupler/controller to be totally dependent on the program "pinned" into the pinboard controller: control may be transferred from either the calculator, PDP11, or the teleprinter to the pinboard controller by attempting to transfer control to an I/O slot that cannot assume control.

Figure 14. Pinboard Controller Program Guide

For example, one might try to transfer control to I/O Slot 1 which is a BCD input card. By giving the @ 1C command, the coupler/controller automatically defaults control to the pinboard controller since the BCD input card cannot assume command. When control is transferred from another source to the pinboard controller and the switch on the front of the pinboard controller is in the *run* position, the program on the pinboard controller will be executed. Again, the pinboard controller is limited to 15 steps or 15 program commands. The pinboard controller can also transfer command back

to the program source which previously transferred control to the pinboard controller by the @ AC command--where A is the slot of the source to which control is transferred and C is the designation to transfer control.

Use of the pinboard controller is somewhat limited with respect to the control that can be obtained from the calculator, PDP11, or the teleprinter tape reader. However, the pinboard controller may be quite useful for executing short (less than 15 program steps) subroutines where the calculator needs to remain for doing other things. The pinboard controller is not very useful in the light of the programming capability of the calculator, however.

Teleprinter Card

The coupler/controller has a teleprinter interface card located in I/O Slot 0 (Figure 5). The teletype printer shown in Figure 3 is a "standard teletype" and can be used to input commands and data to the coupler/controller or to receive information from the coupler/controller and print it. The teleprinter is superior to the calculator as a data recording system because the calculator has only a 3-inch paper tape while the teleprinter has a full 8.5-inch page format, making it possible for information that is reduced and analyzed by the calculator to be output to the teletype for a more normal appearance.

The teleprinter has three general functions (Figure 15):

- As a data receiver,
- As a data source, and
- As a system controller and program source.

As a data receiver, the teleprinter can take five possible commands. Commands require the @ AC where the A is the I/O slot to be addressed and the C is the letter command given to that particular I/O card. Since the teleprinter card is located in I/O Slot 0, all of the commands start out with @ 0 followed by a letter command. A J command causes a carriage return and line feed, the H command causes the teleprinter to print any ASCII characters that follow the H command. The I command causes the teleprinter to print one data word from the source such as the calculator or BCD input card. A K command makes the I command continuously active, and the N command negates the K command.

The teleprinter also can be used as a data source and can cause the coupler/controller to execute commands just as if they were issued from the calculator. The O command causes the teleprinter to output data from the teleprinter to the coupler/controller data

INSTRUCTION	ACTIVITY	TELEPRINTER FUNCTIONS			
		KEYBOARD	PRINTING	TAPE READER	TAPE PUNCH

THE TELEPRINTER AS A SYSTEM CONTROLLER AND PROGRAM SOURCE

@BC (CONTROL)	TAKE CONTROL OF SYSTEM	●	①	●	①
@BD	CAUSES TELEPRINTER TAPE READER TO BE PROGRAM SOURCE	●		●	
@BE	DISABLES THE TAPE READER	●	●		●

THE TELEPRINTER AS A DATA SOURCE

@BO (OUTPUT)	TELEPRINTER INTERRUPTS CONTROLLER AND OUTPUTS DATA TO THE SYSTEM DATA BUS	●		●	
@BH (MANUAL)	ALLOWS THE TELEPRINTER KEYBOARD TO OUTPUT DATA TO THE SYSTEM DATA BUS	●			
@BL (LIST)	CAUSES THE TELEPRINTER TO TRANSFER A PROGRAM (OR DATA) INTO THE SYSTEM	●		●	

THE TELEPRINTER AS A DATA RECEIVER

@BJ	CARRIAGE RETURN AND LINE FEED		●		●
@BI (INPUT)	PRINTS ONE WORD OF DATA FROM A SOURCE		●		●
@BK (CONTINUOUS)	MAKES COMMAND @BI CONTINUOUSLY ACTIVE		●		●
@BN (NEGATE)	NEGATES THE @BK COMMAND				
@BH (HEADING)	TELEPRINTER PRINTS ANY ASCII CHARACTERS FOLLOWING THIS COMMAND		●		●

① PRINTING AND PUNCHING DEPEND ON PREVIOUSLY ISSUED D OR E COMMAND

Figure 15. Teleprinter Program Instructions and Controls

bus. The *O* and *N* commands, as explained in Figure 15, are not often used and will not be discussed further.

When the teleprinter is used as a system controller and program source, three basic commands are employed. The *C* command causes the teleprinter to take control of the entire operating system. The *D* command causes the teleprinter paper tape reader to be used as a program source; that is, when a program and/or data is punched up on paper tape, the paper tape can be read through the teleprinter paper tape reader and the information on the paper tape will be interpreted by the coupler/controller. The *E* command activates the paper tape reader (if on) and also activates the keyboard so characters entered on the keyboard will also appear on the printed paper and be transmitted to the coupler/controller.

The teleprinter can be used in much the same way as the calculator to control the coupler/controller except that the teleprinter cannot interpret data or reduce data, and it has no calculative powers. The teleprinter is simply a data source device, a data receiving device, and a limited controlling device.

Multiprogrammer and Multiprogrammer Extender

The multiprogrammer and multiprogrammer extender are shown in Figures 1 and 2. The multiprogrammer is an output extension of the coupler/controller. The multiprogrammer itself has no input capabilities; that is, it cannot take in data or information from any measuring device and transmit it to any receiver. The multiprogrammer is limited to receiving commands from the coupler/controller, interpreting these commands, and then outputting information to an external environment. Figures 16 and 17 show the exterior of the multiprogrammer with the 16-bit switch register and the interior of the multiprogrammer extender. The output slot allocations in the multiprogrammer and the multiprogrammer extender are shown in Table 4.

The multiprogrammer requires two pieces of information to function. Information is supplied via the 16-bit switch register and may be input to the switch registers either manually or through a remote data source. This discussion will be limited to remote programming of the multiprogrammer.

The *control word* is the first information supplied to the multiprogrammer. The distinctions between the *control word* and the *data word* are shown in Figure 18. Only 11 of the 16 bits are used in the control word. Bits 12, 13, 14, and 15 are all set to indicate control words. Bit 6 is used for a *data transfer enable*, Bit 5 is used for a *system enable*, Bit 4 is used for a *timing mode enable*, and Bit 0 through 3 are used as the *unit address*. Generally only the *system enable* bit or Bit 5 is used,

Figure 16. Multiprogrammer and Multiprogrammer Extender

and Bits 4 and 6 are generally not employed. The *unit address* shown in Bits 0 through 4 designate which main box is being addressed, that is, whether the multiprogrammer or the multiprogrammer extender is being addressed. The multiprogrammer can have up to 15 extender boxes attached to it, but in the case of the ADAS system there is only one. The multiprogrammer unit address is 0 whereas the multiprogrammer extender unit address is 1.

A control word must be used to address an I/O slot in the multiprogrammer extender. The control word is indicated by the presence of Bits 12 through 15, the *system enable* Bit 5, and Bit 0 which indicates a 1 or the *unit address* 1. This tells the multiprogrammer that the next forthcoming data word should be directed to the multiprogrammer extender or Unit 1.

Following the control word there should be a *data* word, a 16-bit word as shown in Figure 18. (The data word need not immediately follow the control word. The previous control word will always be assumed until a new control word is issued.) Bits 12, 13, 14, and 15 indicate the slot address. Each of the boxes (the multiprogrammer and the multiprogrammer extender) has 14 slots. The

Figure 17. Interiors of Multiprogrammer and Multiprogrammer Extender

slot address can go from 0000 to 1110 (binary). The remaining 12 bits of the control word are interpreted by the particular output slot addressed. Once the 12-bit data word is interpreted by the output card; the desired output from the card is available at the cards's output terminals.

For instance, the following commands would be used to address the digital-to-analog (D/A) converter cards located in Multiprogrammer Slots 400 through 403: 1111000000100000 (control word) and 0011010000000000 (data word).

The control word would enable the system and direct the data word to Unit Address 0 (the multiprogrammer). The data word addresses Slot 3 which contains a digital-to-analog converter. The following 12-bits of data indicate a 5.120 volt output.

Subroutines which the user can call from the mainline program, have been written for each of these cards. The subroutines will figure out the control and data words.

The D/A converter card interprets digital data and converts it to an analog voltage. Any voltage between -10.235 and +10.235 may

Table 4. Multiprogrammer and Multiprogrammer
Extender Output Slot Allocation

Output Slot Number	Allocation
<hr/> HP6936A Multiprogrammer <hr/>	
400-403	D/A Converter (12-bit-volt) (Number 69321-60022)
404-405	Empty
406-407	TTL Digital Outputs (Number 69331-073)
408	Empty
409-411	Multi-Relay Cards (Number 69330-60021)
<hr/> HP6941B Multiprogrammer Extender <hr/>	
400-411	Multi-Relay Cards (Number 69330-60021)

be obtained in steps of 5 millivolts. Refer to the appendices for the subroutine which converts the desired voltage into the corresponding data word.

The TTL digital output cards contain 12 bits. Through the proper control word, any of these 12 bits can be set to 1 or 0. Output levels are the standard TTL levels: 0 and 5 volts respectively for 0s and 1s. The TTL output card can sink 32 milliamperes in its low state and has a 1 kilohm source impedance for sourcing in its high state.

The multirelay cards are simple relay closures located on the relay card. There are 12 such relays located on each card. They can be used simply as open or closed switches and are programmed directly by the 12 data bits in the control word.

All the output terminals for the output cards located in either the multiprogrammer or the multiprogrammer extender are brought out to the digital and analog touch panels as shown in Figures 8 and 9.

Figure 18. Multiprogrammer Word Formats

Figure 19 provides a simple example of the use of the multirelay card located in the multiprogrammer extender. Bits 15, 13, 12, 6 and 1 have been set on the 16-bit switch register at the bottom of the figure. Assuming that the *control* word has already been given and what is present on the switch register is the *data* word, this *data* word would be interpreted as follows: Bits 15, 13, and 12 are the slot address and are interpreted as Slot 11 (in a standard BCD code). (Bit 15 is a 4; 13 is a 2; and 12 is a 1--the sum of which is 11.) Bits 10, 6, 3, and 0 have been engaged, which closes Relays 0 3, 6, and 10 on the relay card located in Slot 11 of the multiprogrammer extender.

Figure 19. Example of Relay Card in Multiprogrammer Extender

LED lights have been mounted on the end of the far right card connector immediately above the open door of the multiprogrammer extender. When a switch or a relay has been closed, the appropriate light will come on. Counting from the top, Lights 0, 3, 7, and 10 are turned on. Therefore the relays corresponding to the pattern designated by the switch register on the multiprogrammers have been engaged.

The multiprogrammer is strictly an output oriented device. It is not capable of inputting information from any external source and transmitting that information to the coupler/controller. Information in the form of *control* and *data* words comes from the BCD output card located in the coupler/controller I/O Slot 3.

Use of a BCD output card to program the 16-bit information required by the multiprogrammer is inefficient because only 16 of the 40 possible bits available on the BCD output card are used. When an output string in decimal is transmitted from the calculator through the BCD output card to the multiprogrammer, it is interpreted from left to right or from the most significant to the least significant digit.

Once the decimal number is changed to a BCD number, only the last two bits, or the most right-hand justified BCD bits, are interpreted by the multiprogrammer. Therefore, the largest decimal number that can be output with any meaning to the multiprogrammer via the BCD output card is a decimal 3. A decimal 3 would initiate or light two adjacent switch registers. For instance, to obtain the control word where Bits 15, 14, 13, and 12 are initiated, one would output the decimal combination of 33. The first 3 would initiate Bits 15 and 14 and the second 3 would initiate Bits 13 and 12. It can be seen from this example that each decimal number controls 2 binary bits on the 16-bit switch register. To address Slot 11 as shown in Figure 19, the decimal would be 23, the 2 would initiate Bit 14, and the 3 would initiate Bits 13 and 12. The remainder of that data word in decimal would be 101021 which corresponds to the light pattern shown in Figure 19.

It is not absolutely necessary for the operator of this system to understand completely the coding format used in the multiprogrammer and the multiprogrammer extender. He may simply refer to the subroutines shown in the appendices and explained in greater detail in the systems operating manual.

Sample Application: Tantalum Nitride Resistor Life Test

One of the original justifications for the purchase of the ADAS system was to set up an automatic system to measure life stability on thin film tantalum nitride resistors. The software program shown in Figures 20 through 24 was the first attempt to program ADAS for automatic data acquisition and data reduction.

The general formula for figuring the percent change in resistance overtime is shown below.

$$\Delta R/R = \frac{R(X+20) - R(X)}{R(X)} \bigg|_{X=1}^{12} * 100$$

The life test program data is summarized in Tables 5 through 8. Table 5 provides pin data on the 12 individual resistors while Table 6 provides data register allocations. Table 7 summarizes measurement times and relay addresses and Table 8 gives the individual data file numbers for the test.

The tantalum nitride resistor life test apparatus uses the Statham oven, the multiprogrammer and multiprogrammer extender, the coupler/controller, and the HP9820 calculator shown in Figure 1. The Statham oven has been fixtured to receive 12 ceramic substrates. Each ceramic substrate contains 12 thin film resistors. The resistor values are shown in Table 5. The purpose of the test is to measure the initial value of the resistors and then to age the

Table 5. Pin Data for 12 Individual Resistors on TaN Resistor Life Test Program

Resistor	Value (Ohms)	High Pin	Common Pin
1	20	22	47
2	50	3	
3	100	5	
4	200	7	
5	500	9	
6	1000	11	
7	2000	13	
8	5000	15	
9	10,000	17	
10	20,000	19	
11	50,000	20	
12	100,000	1	47

resistors at 150°C for 1000 hours. During the 1000 hour aging period the resistors are again measured at the 2, 6, 10, 40, 100, 200, and 600 hour periods. The data from each incremental time measurement is applied to the general formula and the change in resistance is calculated.

The major portion of the tantalum nitride resistor life test program as charted in Figures 20 through 24 is concerned with generating the relay address. Each of the 144 resistors in the test is wired to an individual relay in the multiprogrammer extender. To measure each resistor, an individual relay must be closed to connect the resistor to the input terminals of the digital multimeter. When told to do so by the BCD input card, the digital multimeter takes a measurement of the resistance value and transfers that measurement to the BCD input card in the coupler/controller.

Figure 20 shows the main line computer program flow chart for the life test. It can be seen that the main line simply requests the entering of particular information such as the data and oven

Table 6. Data Register Allocation
for TaN Life Test

Data Register Allocation

R(0) => Clock Time
R(1)-R(12) => Initial Resistor Values
In Files 2-13
R(13-20) => $\Delta R/R$ for Each Substrate
R(21)-R(32) => Measured Value of
Resistors at Measuring Time
(Files 14-109)
R35 => Time-Test Number
R(41)-R(52) => Σ of Individual
Resistor Values for Each Value
Resistor During *INTL*
R(36)-R(37) => Σ of $\frac{\Delta R}{R}$ during *List*

temperature for the start of the program and then calls subroutines to scan the resistors through relay closures. There are six basic subroutines to the main line program:

- *Date* transmits the day, month, and year through the coupler/controller and prints it on the teletype;
- *Time* interrogates clock/timer/pacer card for the proper time and records it on the teletype;
- *Scan* scans the relays and measures the resistor values;
- *List* prints the resistor values on the teletype;
- *INTL* records the initial values of the resistors at time 0; and
- *Plot* plots the percent change in resistance on a semilog scale over 1000 hours.

Flow charts for these subroutines are shown in Figures 20 through 24. The actual program is listed in Appendix F.

Table 7. Measurement Times and Relay Addresses For TaN Resistor Life Test

Measurement Times (Hours)	Relay Address	Card Slot Number
0	XX:00	XX000001
2	XX:01	XX000002
6	XX:02	XX000010
10	XX:03	XX000020
40	XX:10	XX000100
100	XX:11	XX000200
200	XX:12	XX001000
600	XX:13	XX002000
1000	XX:20	XX010000
	XX:21	XX020000
	XX:22	XX100000
	XX:23	XX200000

ACCOMPLISHMENTS

The Electronics Laboratory ADAS system has been completed. Approximately 200 software programs have been written for the HP9820 calculator. Of these, approximately 50 have been designed to interface with the coupler/controller and gather laboratory information. The ADAS system has been very successful in reducing the burden on engineer and technician workloads.

The automation of the resistor life test program alone is estimated to have saved approximately 1 manyear per year. The highly flexible input/output format of the ADAS system has permitted many other laboratory instruments to be adapted to the ADAS system.

The choice of the HP9820 calculator and HP2570 coupler/controller has proven to be the best for a laboratory-based ADAS system. There has been virtually no down time on any of the hardware components in the ADAS system.

Table 8. Individual Data File Numbers for TaN Resistor Life Test

Test Number	1	2	3	4	5	6	7	8	9
R(35)	0	1	2	3	4	5	6	7	8
Time (Hours)	0	2	6	10	50	100	200	582	1000
Fixture Number	Individual Data File Numbers								
1	2	14	26	38	50	62	74	86	98
2	3	15	27	39	51	63	75	87	99
3	4	16	28	40	52	64	76	88	100
4	5	17	29	41	53	65	77	89	101
5	6	18	30	42	54	66	78	90	102
6	7	19	31	43	55	67	79	91	103
7	8	20	32	44	56	68	80	92	104
8	9	21	33	45	57	69	81	93	105
9	10	22	34	46	58	70	82	94	106
10	11	23	35	47	59	71	83	95	107
11	12	24	36	48	60	72	84	96	108
12	13	25	37	49	61	73	85	97	109

FUTURE WORK

It is recommended that the manual patch panels shown in Figure 10 be replaced with clip-on type patch panels to allow the operator to replace whole patch panel interconnections in a single step rather than plugging in individual patch panel cables. This would not necessarily improve the ADAS in a technical sense but would decrease setup time.

Considerable future work needs to be done in terms of software programming the ADAS system to accomplish new jobs in the laboratory. Most future work probably will be based on a software format. The tremendous flexibility of the ADAS system allows for quickly developing software routines as opposed to developing new hardware measurement apparatus.

Figure 20. TaN Resistor Life Test Mainline Flow Chart

Figure 21. TaN Resistor Life Test *Scan* Subroutine Flow Chart

Figure 22. TaN Resistor Life Test *INTL* Subroutine Flow Chart

Figure 23. TaN Resistor Life Test *List* Subroutine Flow Chart

Figure 24. TaN Resistor
Life Test *Time*
Subroutine Flow
Chart

Appendix A

DMM PATCHBOARD PROGRAMMING HP3490A BCD OUTPUT TO HP2570A SLOT 4 BCD INPUT CARD

From	To	Comments
C1 (Data Flag)	P18	
C2 (Hold)	P20 & C18	Ground
C3 (Ext Encd)	P19	
C5 (High Ref)	P16	
C6 (Low Ref)	P15	
C7 (Print Hold)	P13	
B1	D1	
B2	D2	
B3	D3	Location 5
B4	D4	Most Significant
B8	C17	Location 4
A2	C14	Over Range
A3	C15	Over C
A4	C16	Over B
A5	D17	Over A
A6	D18	
A7	D19	Location 6
A8	D20	
A9	D13	
A10	D14	
A11	D15	Location 7
A12	D16	
A13	D9	
A14	D10	
A15	D11	Location 8
A16	D12	
A17	D5	
A18	D6	
A19	D7	Location 9
A20	D8	Least Significant

Appendix B

PATCH PANEL AND REAR CONNECTOR WIRE LIST

Front Panel Left Patchboard	Rear Panel Connectors
LA1-LA20	JT1-JT20
LB1-LB20	JT21-JT40
LC1-LC20	JY1-JY20
LD1-LD20	JY21-JY40
LE1-LE20	JJ1-JJ20
LF1-LF8	JJ21-JJ28
LF9-LF20	JE1-JE12
LG1-LG20	JF1-JF20
LH1-LH20	JG1-JG20
LI1-LI20	JH1-JH20
LJ1-LJ20	JV1-JV20
LK1-LK20	JV21-JV40
LL1-LL20	JX1-JX20
LM1-LM20	JX21-JX40
LN1-LN20	JI1-JI20
LP1-LP8	JV41-JV48
LP9-LP12	JF21-JF24
LP13-LP20	JT41-JT48
Front Panel Right Switchboard	Rear Panel Connectors
RA1-RA20	JZ1-JZ20
RB1-RB20	JZ21-JZ40
RC1-RC20	JA1-JA20
RD1-RD20	JA21-JA40
RE1-RE12	JN1-JN12
RE13-RE14	JB1-JB2
RE15-RE16	JC1-JC2
RE17-RE20	JD1-JD4
RF1-RF12	JN13-JN24
RF13-RF14	JB3-JB4
RF15-RF16	JC3-JC4
RF17-RF20	JD5-JD8
RG1-RG12	JK1-JK12
RG13-RG18	JM1-JM6
RG19-RG20*	No Connection

Front Panel Right Switchboard	Rear Panel Connectors
RH1-RH12	JK13-JK24
RH13-RH18	JM7-JM12
RH19-RH20*	No Connection
RI1-RI12	JL1-JL12
RI13-RI18	JM13-JM18
RI19-RI20**	No Connection
RJ1-RJ12	JL13-JL24
RJ13-RJ18	JM19-JM24
RJ19-RJ20**	No Connection
RK1-RK20	JW1-JW20
RL1-RL20	JW21-JW40
RM1-RM20	JP1-JP20
RN1-RN20	JR1-JR20
RP1-RP18	JS1-JS18
RP19-RP20	Chasis Ground

Front Panel Patchboard	Rear Panel Connectors
No Connection	JA41-JA50
	JB5-JB14
	JC5-JC14
	JD9-JD14
	JE13-JE14
	JF25-JF36
	JG21-JG36
	JH21-JH36
	JI21-JI36
	JJ29-JJ36
	JK25-JK36
	JL25-JL36
	JM25-JM36
	JN25-JN36
	JP21-JP36
	JR21-JR36
	JS19-JS36
	JT49-JT50
	JV49-JV50
No Connection	JW41-JW50

Front Panel Patchboard	Rear Panel Connectors
No Connection	JX41-JX50
No Connection	JY41-JY50
No Connection	JZ41-JZ50
*RG19, RG20, RH19, and RH20 all tied together	
**RI19, RI20, RJ19, and RJ20 all tied together	

Appendix C

FUNCTIONAL INPUTS TO REAR CONNECTORS OF PATCH PANEL ASSEMBLY

Connection Identification	Number of Pins	Function Inputs
A	50	"B" Inputs
B	14	Power Supplies
C	14	DMM Analog Inputs (HP3490A Rear Panel)
D	14	D/A Converters (HP Multiprogrammer HP6937A)
E	14	DMM Remote Control (HP3490A-BCD Input)
F	36	Digital Micrometer (IKL 1338-B)
G	36	Counter Remote Control (HP5325B)
H	36	Pulse Generator Remote Control (Tektronic R116)
I	36	Digital Spare
J	36	Counter Data Output (HP5325B)
K	36	Multi-Relay Card (HP6937A Slot 10)
L	36	Multi-Relay Card (HP6937A Slot 11)
M	36	Ten Channel Scanner (HP2570A)
N	36	Multi-Relay Card (HP6937A Slot 9)
P	36	
R	36	
S	36	
T	50	BCD Input (HP2570A Slot 4)
V	50	BCD Input (HP2570A Slot 1)
W	50	"C" Inputs
X	50	BCD Output (HP2570A Slot 2)
Y	50	Digital Multimeter BCD Data Output (HP3490A)
Z	50	"A" Inputs

Appendix D

DMM REMOTE CONTROL PATCHBOARD PROGRAMMING HP3490 DMM REMOTE CONTROL FROM BCD (NEGATIVE TRUE LOGIC) OUTPUT SLOT 2 (FORMAT: "@2I1FAR1@4E")

From	To	Comments
F9	L4	Remote Enable
F10	L16	Range A
F11	L15	Range B
F12	L14	Range C
F13	L12	Autorange
F14	L8	Function A
F15	L7	Function B
F16	L20	Program Execute

To program the HP3490 DMM, use the following format statement:

FMT "@2I1FAR1@4E"

The 1 Preceding the FAR is the Remote Enable
The 1 Following the FAR is the Program Execute

F => Function: F = 0 => Self Test Mode
 F = 1 => AC Volts
 F = 2 => Ohms
 F = 3 => DC Volts

A => Autorange: Even # => Autorange
 Odd # => Non-autorange

R => Range: R = 1 => 0.1 V or 0.1 kV
 R = 2 => 1V or 1 kV
 R = 3 => 10 V or 10 kV
 R = 4 => 100 V or 100 kV
 R = 5 => 1000 V or 1000 kV
 R = 6 => 10,000 kΩ

Appendix E

RELAY, DRVR, CONVERT, CODE AND CONVERT, VOLT, AND CLOCK READ SUBROUTINES

RELAY SUBROUTINE

Enter GSB Relay
R0 = Slots 0-11
R1 = Relays 1-12
Negative - Open
Positive - Close

```
10:
"RELAY";ABS R1→R
2:GTO +7;IF 0≤R0
;IF R0≤11;GTO +8
;IF 1≤R2;IF R2≤1
2:JMP 1F
11:
INT ((R2-1)/2)→R
3:R2-2R3→R2;R(6+
R0)/10→R3→R4F
12:
R4-10INT (R4/10)
→R4;R2+10→R3→R3;
INT (R4/R2)→R2;
IF R2=2INT (R2/2
);JMP 2F
13:
-R3→R3;-R1→R1F
14:
GTO +2;IF 0≤R1;R
3→R(6+R0)→R1→R(6
+R0);JMP 1F
15:
(R0-4INT (R0/4)+
10INT (R0/4))*10
→6→R5;R5+R1→R1;
RET F
16:
PRT "COMND ERR";
JMP 3F
17:
PRT "SLOT ERR";
JMP 2F
18:
PRT "RELAY ERR" F
19:
GTO 3F
20:
END F
```

$\frac{<ABS(R1)<12}{0<R0<11}$

OK TO ADD: +2

OK TO SUB: CHANGE TO NEG:
REVERSE IND
IF IND SAYS ADD - ADD

COMPUTE SLOT ADDRESS AND MERGE
WITH DATA.

RET*

ERROR ROUTINES

*NOTE: CHANGE TO R5+R1→R0;
1→R1; GSB "DRVR" TO OUTPUT
COMM BEFORE RETURN.

RETURNS COMMAND IN R1; SLOT
ADDRESS IN R5.

RELAY CALLING PROGRAM

This routine is similar to the Code and Conv subroutines. Storage Registers 6 through 17 are allotted for use as storage: one register per relay card.

This program merely checks the accuracy of the subroutine.

0:	ZERO COUNTER
PRT " EXAMPLE US	
E OF", "*** RELA	ZERO ALL REGISTERS 0-17
Y ***" SPC 2H	
1:	INPUT SLOT #; (0-11)
FXD 0;0+AH	RELAY #; (1-12) NEG FOR OPEN
2:	POS FOR CLOSE
0+RA;JMP (A+1+R)	
=17H	PRINT INPUTS
3:	
ENT "SLOT =?",R0	UPDATE CODE IF OPERATION IS
,"RELAY ?",R1H	LEGAL
4:	
PRT R0,R1H	PRINT COMMAND WORD RETURNED
5:	
GSB "RELAY" H	
6:	
PRT R1;GTO 3H	
7:	
0+ZH	
8:	
0+ZH	
9:	
0+ZH	

NOTE: Relay subroutine keeps actual output command last given for each slot.

Only the digit corresponding to the relay in question is checked and updated, thus saving time used by base conversion algorithm.

SINGLE RELAY CLOSURE

Subroutines to take Relay Number (1 to 144) and compute command data to send to 2570A to 6940A multiprogrammer.

This example is broken down into a number of small subroutines to make it easier to follow. That is a good process to follow in developing programs. It makes for easy debugging. After the routine works as expected, it is fairly simple to compact instructions for time and memory saving.

0:	KEEPS COUNTER IN A REGULAR
PRT "DEMO OF SIN	COUNTER SET IN R0 RELAY SELECTOR -
GLE", "RELAY SELE	PRINT # OF RELAY SELECT- "COMWD"
CT SUBS", "REVISI	COMPUTES OUTPUT COMMAND WORD "CL"
ON 2" F	SIMULATES OUTPUT SUBROUTINE
1:	
FWD 0:0+R5;1+R1	R4=NEW SLOT # R5 = OLD SLOT #
2:	IF DIFFERENT. OLD SLOT MUST BE
A+R0;PRT R0 F	ISSUED CLEAR (ALL ZEROS)
3:	
GSB "COMWD" F	SET POINTER TO NEW COMMAND WORD
4:	AND ISSUE.
GSB "CL" F	
5:	SAVE NEW SLOT # IN OLD - FOR
A+1+R1;GTO 2 F	NEXT TIME-RET.
6:	
"CL";IF R4=R5;5+	ISSUE COMMAND WORD POINTED TO
R1;GSB "CO" F	BY R1 TO SYSTEM (IN THIS CASE -
7:	PRINT) - RET
0+R1;GSB "CO" F	
8:	
R4+R5;RET F	
9:	
"CO";PRT RR1;	
RET F	
10:	
0+Z F	
11:	
0+Z F	

NOTE: CL and CO are simulate routines to print data instead of transmitting it to system.

NOTE: Assuming 12 relay cards in Slots 0-11, each having 12 continuous closures numbers top to bottom left to right.

R5 Register is used here to keep track of which slot number was addressed last. Another register could be used by this and other routines to keep track of which box (6936 or 6937) (Unit Number 0 or 1) was last addressed.

```

12:
"CLEAR":IF R4=R5
:5+R1:GSB "CMOUT"
"┐
13:
0+R1:GSB "CMOUT"
"┐
14:
R4+R5:RET ┐
15:
"CMOUT":FMT "06I
07F",FXD 9.0,"",
,Z:WRT 4,R1:
RET ┐
16:
"COMWD":GTO "ERR
":IF R0>0:IF 145
>R0:R0-1+R0:GSB
"DCODE"┐
17:
GSB "RNUM"┐
18:
GSB "SLNUM"┐
19:
R3+R4+R0:RET ┐
20:
"DCODE":INT (R0/
12)+R2:1+R0-12R2
+R1:RET ┐
21:
"RNUM":(2-(R1-2
INT (R1/2)))*10+
INT ((R1-1)/2)+R
3:RET ┐
22:
"SLNUM":(R2-4
INT (R2/4)+10
INT (R2/4))*10+6
+R4:RET ┐
23:
"ERR":DSP "ERROR
":PRT "NO SUCH R
ELAY":STP ┐
24:
GTO 0┐
25:
END ┐
R344

```

SHOULD CONTAIN UNIT SELECT
COMMAND SUCH AS <3301209,>
INSERTED BETWEEN @7F AND END
QUOTE.

ENT:R0=RELAY#

CHECK LIMITS OF RELAY #
(0<#<145)

DECREMENT RELAY # (NOW IN RANGE
0 TO 143) BREAK THAT INTO SLOT #
AND CONTACT #

CONVERT CONTACT # TO BASE 4

CONVERT SLOT # TO BASE 4

COMBINE AND RETURN R0=COMMAND WORD
R4=SLOT #

ENT:R0=RELAY#(0-143)

RET:R1=CONTACT#(1-12)

R2=SLOT#(0-11)

ENT:R1=CONTACT#(1-12)

RET:R3=C#BASE 4 (00001-20000)

ENT:R2=SLOT#(1-12)

RET:R4=S#BASE 4 E6 (00000000 -
30000000)

SINGLE CONTACT CLOSURE VERSION 2

```

0:
PRT "DEMO OF SIN
GLE"; "RELAY SELE
CT SUBS"; "VERSIO
N 3" F
1:
FXD 0; 0 R3; 1 A F
2:
A R0; PRT R0 F
3:
GSB "COMWD" F
4:
A+1 A; GTO 2 F
5:
0 Z F
6:
0 Z F
7:
"COMWD"; GTO +6;
IF R0 > 0; IF 145 > R
0; JMP 1 F
8:
INT ((R0-1)/12) +
R2; R0-12R2+R1; (2
-(R1-2INT (R1/2)
)) * 10 + INT ((R1-1
)/2) + R0 F
9:
(R2-4INT (R2/4) +
10INT (R2/4)) * 10
+ 6 + R1; IF R1 = R3; R
3 + R2; R1 = R3; GSB "
OUT" F
10:
R0 + R1 + R2; GSB "OU
T" F
11:
RET F
12:
"OUT"; PRT R2;
RET F
13:
PRT "NO SUCH REL
AY"; STP F
14:
GTO 0 F
15:
END F
R371

```

This is a compacted version of same routine.

It is more difficult to follow but it saves space. It is a good policy to save Version 1 routines on a separate tape for possible use in further development. It is also recommended that final versions be moved to a separate tape to prevent loss during program development.

OUT would be replaced with DRV R for command of coupler, multiprogrammer.

RELAY NUMBERING FOR SINGLE RELAY SELECT

SLOT	0	1	2	3	4	5	6	7	8	9	10	11
1	13	25	37	49	61	73	85	97	109	121	133	
2	14	26	38	50	62	74	86	98	110	122	134	
3	15	27	39	51	63	75	87	99	111	123	135	
4	16	28	40	52	64	76	88	100	112	124	136	
5	17	29	41	53	65	77	89	101	113	125	137	
6	18	30	42	54	66	78	90	102	114	126	138	
7	19	31	43	55	67	79	91	103	115	127	139	
8	20	32	44	56	68	80	92	104	116	128	140	
9	21	33	45	57	69	81	93	105	117	129	141	
10	22	34	46	58	70	82	94	106	118	130	142	
11	23	35	47	59	71	83	95	107	119	131	143	
12	24	36	48	60	72	84	96	108	120	132	144	

DRVR SUBROUTINE

```

0:
"DRVR":R1+330120
0+R1:IF R1=RX:1+
RX:GSB +3+
1:
0+RX:GSB +2+
2:
R1+RX:RET +
3:
FMT "06107F":
FXD 9.0:"":Z:
WRT 4:RRX+Z:RET
+
R404

```

Entry: GSB "DRVR"

R0 = Command Word
R1 = Unit # (0 or 1)

NOTE: RX is some register saved for
exclusive use of DRVR

This allows program to call one DRVR program
to control both 6936 and 6937 without
keeping track of sequence of calls.

CONVERT SUBROUTINE

Convert any number to digital notation to any numeral base.

```
0:
ENT "BASE= ?";R0
;0→A;FXD 2F
1:
A+1→A+R1;GSB 'CO
NV"F
2:
PRT A,R3;SPC 1;
GTO 1F
3:
"CONV";0→R3F
4:
INT (LOG R1/LOG
R0)→R2;10↑R2+R3→
R3;R1-R0↑R2→R1;
GTO +0;IF 1>R1;
RET F
5:
END F
R399
```

CODE AND CONVERT CALLING PROGRAM

Uses R6-R17 for storing data pattern codes for each relay card.

```
0:
PRT "EXAMPLE USE
  OF", "CODE AND C
  ONV"; SPC 2F
1:
FXD 0; 0+AF
2:
0+RA; JMP (A+1+A)
=17F
3:
ENT "SLOT =?", R0
, "RELAY ?", R1F
4:
PRT R0, R1F
5:
GSB "CODE" F
6:
R1+R; GSB "CONV" F
7:
PRT R1; GTO 3F
8:
0+ZF
9:
0+ZF
10:
0+ZF
```

RESET COUNTER (A)

ZERØ ALL REGISTERS 0-17

SLOT # (0-11)

RELAY # (1-12)

(NEGATIVE TO OPEN)

(POS TO CLOSE)

PRINTS ENTRIES

UPDATE STORED VALUE FOR THAT SLOT

CHECK IF OPERATION VALID

CONVERT STORED CODE (UPDATED)

TO OUTPUT COMMAND FORMAT

PRINT RESULTS INSTEAD OF OUTPUTTING

CODE AND CONVERT LISTINGS

```

11:
"CODE";R(6+R0)÷R
2:2↑ABS R1/2÷R3;
12÷R4;IF R1=0;-R
2÷R3;JMP 3F
12:
R4-1÷R4;IF 0>R4;
-R1÷R1;JMP 2F
13:
2↑R4÷R5;GTO -1;
IF R5<R2;R2-R5÷R
2;IF R5=R3;-R3÷R
3;JMP 1F
14:
IF R1<0;INT (R(6
+R0)+R3)÷R1÷R(6+
R0);RET F
15:
PRT "CODE ERR";
STP F
16:
GTO 3F
17:
"CONV";GTO +3;
IF 0<R0;IF R0<11
;IF R1<4095;IF 0
<R1;GSB "BAS4"
18:
R3÷R5;R0÷R1;GSB
"BAS4"
19:
R3*10+6+R5÷R1;
RET F
20:
PRT "CONV ERR";
STP F
21:
GTO 3F
22:
"BAS4";0÷R3;LOG
4÷R4;IF R1=0;
RET F
23:
INT (LOG R1/R4)÷
R2;10↑R2+R3÷R3;R
1-4↑R2÷R1;IF R1>
0;JMP 0F
24:
RET F
25:
END F

```

ENT: R0 = SLOT # (0-11)
R1 = RELAY # (1-12) - FOR OPEN
+ FOR CLOSE
0 FOR RESET

CHECKS FOR VALID OPERATION
TRYING TO OPEN RELAY NOT CLOSED -
"CODE ERR"
TRYING TO CLOSE RELAY ALREADY
CLOSED - "CODE ERR"
VALID OPERATION UPDATES STORAGE
AND RETURNS WITH NEW VALUE IN R1.
R0 STILL = SLOT #

ENT: R0 = SLOT # (0-11)
R1 = VALUE (NEW) FOR THAT
SLOT
CHECKS FOR NEG SLOT # --"CONV ERR"
CHECKS FOR SLOT # < 11 -- "CONV ERR"
CHECKS FOR NEG VALUE (ILLEGAL) --
"CONV ERR"
CONVERTS VALUE TO OUTPUT FORMAT
READY TO BE SENT TO COUPLER.
RETURNS COMMAND WORD IN R1

VOLT SUBROUTINE

ENT R0= SLOT # OF ADC (0-3)
 R1= VOLTAGE VALUE (IN VOLTS)
 $-10.24 \leq R1 \leq 10.235$

RETURNS COMMAND WORD IN R1

SHOULD BE 3 FOR SLOT 0-3

0:	12:
FXD 3;ENT "VOLTA	GTO +4;IF 0<R0;
GE =?";R1;"SLOT	IF R0<11;(R0-4
=?";R0+	INT (R0/4)+10
1:	INT (R0/4))*10+6
PRT R0,R1;GSB "V	+R5+R1;RET +
OLT"+	13:
2:	0+Z+
PRT R1;SPC 1;	14:
GTO 0+	0+Z+
3:	15:
0+Z+	PRT "RANGE ERR";
4:	JMP 2+
0+Z+	16:
5:	PRT "SLOT ERR"+
0+Z+	17:
6:	SPC 2;STP +
0+Z+	18:
7:	GTO 0+
0+Z+	19:
8:	END +
0+Z+	R367
9:	
"VOLT";0+R4+R5;	
IF 0>R1;R1+10.24	
+R1;2E5+R5+	
10:	
GTO +5;IF 0<R1;	
IF R1<10.235;	
INT (R1/.005)+R1	
;JMP 1+	
11:	
R1/4+R2;INT R2+R	
1;(R2-R1)*4*10+R	
4+R5+R5;R4+1+R4;	
IF R1>0;JMP 0+	

CLOCK READ AND PRINT

```

0:
FMT "@7007I07K06
0";Z;WRT 4;RED 4
;A;B;C;X;FMT "@7
N";Z;WRT 4;PRT "
DAYS:";A;SPC 1F
1:
PRT "HOURS:";B;
SPC 1;PRT "MINUT
ES:";C;SPC 1;
PRT "SECONDS:";X
;SPC 8F
2:
END F
R402

```

```

DAYS:
111.00

HOURS:
11.00

MINUTES:
14.00

SECONDS:
14.00

```

```

THIS PROGRAM
CONVERTS THE
DECIMAL NO. B
INTO TERTIARY NO
C (BASE-4)

```

```

0:
"CONV";0+CF
1:
INT (LOG B/LOG 4
)+R0;10+R0+C+C;B
-4+R0+B;GTO +0;
IF 1>B;RET F
2:
END F
R409

```


Appendix F

RESISTOR LIFE STABILITY TEST SUBROUTINE

MAINLINE

```

0:  ENT "MONTH ?";C;      11:  GSB "SCAN"         26:  IF C=7;FMT "@@HJ
ENT "DAY ?";X;          12:  IF R35=0;JMP 3F    ULY ",Z;WRT 4;
ENT "YEAR ?";Y;        13:  GSB "LIST"         GTO "DAY"
ENT "NO. OF RESI      14:  JMP 3F          27:  IF C=8;FMT "@UGU
STORS";A;              15:  GSB "INTL"        ST ",Z;WRT 4;
1:  ENT "OVEN TEMPER?  16:  GTO "STR"         GTO "DAY"
";B;ENT "TEST;NU      17:  IF R35=8;JMP 3F    28:  IF C=9;FMT "@@HS
MBER ?";R35;R35-      18:  GTO "STR"        EPTMBER ",Z;
2+R35;ENT "TIME      19:  "DATE";FMT "@@J@   WRT 4;GTO "DAY"
?";R0;                20:  IF C=1;FMT "@@HJ    29:  IF C=10;FMT "@@H
2:  FMT "@@J@@H        21:  EBRUARY ",Z;      OCTOBER ",Z;
1 000 HOUR LIFE TE   22:  IF C=3;FMT "@@HM    WRT 4;GTO "DAY"
ST";Z;WRT 4;GSB      23:  IF C=4;FMT "@@HA    30:  IF C=11;FMT "@@H
"DATE"               24:  PRIL ",Z;WRT 4;   NOVEMBER ",Z;
3:  FMT "@@J@@J@@J@   25:  IF C=5;FMT "@@HM    WRT 4;GTO "DAY"
J@@J@@H @0I@70      26:  IF C=6;FMT "@@HJ    31:  IF C=12;FMT "@@H
";FXD 3.0,"",",Z;    27:  UNE ",Z;WRT 4;   DECEMBER ",Z;
WRT 4,R0;            28:  GTO "DAY"        32:  "DAY";FMT "@@I@7
4:  FMT "@@H - HOUR   29:  0";FXD 4.0,"",",Z;   WRT 4,X,Y;
DATA",Z;WRT 4;       30:  RET
5:  FMT "@@J@@J@@H    31:
OVEN TEMPERAT        32:
URE = @0I@70";       33:
FXD 3.0,"",",Z;      34:
WRT 4,B;             35:
6:  FMT "@@J@@J@@H    36:
TEST CONTAINS        37:
@0I@70";FXD 3.      38:
0,"",",Z;WRT 4,A;    39:
7:  FMT "@@H RESIST   40:
ORS@0J@@J";Z;        41:
WRT 4;               42:
8:  IF R35=-1;JMP 2F  43:
9:  LDF 1,R33;        44:
10: GSB "TIME"        45:

```

SUBROUTINE TIME

```

34:
"TIME":FMT "@0J0
0J00H      CLOCK
TIME:  @6L00I070
06M";Z;WRT 4;0+X
F
35:
X+1+X;IF X<130:
JMP 0F
36:
0+X;RET F

```

SUBROUTINE INTL

```

54:
"INTL":0+A+B+C:1
+Y;40+XF
55:
X+1+X;0+RX;IF X<
52;JMP 0F
56:
0+X;Y+1+Y;FMT "@
0J00J00J00H  IN
ITIAL RESISTOR V
ALUES";Z;WRT 4F
57:
FXD 0;FMT "@0H -
POSITION 00I070
";FXD 2;Z;WRT 4;
Y-1;LDF Y,R1,R12
F
58:
X+1+X;FMT "@0J00
H      ";Z;WRT 4F
59:
FXD 4;FMT "@0I07
0";FXD 11;Z;WRT
4;RXF
60:
RX+R(X+40)+R(X+4
0);IF X<11;JMP -
2F
61:
IF Y<12;JMP -5F
62:
0+X;FMT "@0J00J0
0J00H      AVERAG
E RESISTOR VALUE
S -EACH SIZE";Z;
WRT 4F
63:
X+1+X;R(X+40)/12
+R(X+40);FMT "@0
J00H      00I070"
,FXD 11;Z;WRT 4;
R(X+40)F
64:
IF X<11;JMP -1F
65:
RET F

```

SUBROUTINE SCAN

```

37: "SCAN";33000201+
A;FMT "@3I07F",
FXD 9.0,"",Z;
WRT 4,A+
38: -1+B+C+X;-010000
00+A;0+Y;R35+1+R
35+
39: "Y-COUNT";Y+1+Y;
-1+B+C+X;SPC 5;
FMT "@7A",Z;WRT.
4+
40: IF Y=5;A+7E6+A;
JMP 3+
41: IF Y=9;A+7E6+A;
JMP 2+
42: A+1E6+A+
43: "X-INCRS";X+1+X+
44: IF X=0;A+1+A;
JMP 3+
45: IF X=INT (X/2)*2
;B+1+B;A+8*10+B+
A;JMP 2+
46: C+1+C;A+10+C+A+
47: FMT "@3I07F",
FXD 9.0,"",Z;
WRT 4,A;FXD 4+
48: FMT "@4E07I040",
Z;WRT 4;RED 4,R(
X+1);PRT 1000R(X
+1)+R(X+1)+
49: IF X=11;FMT "@3I
07F",FXD 9.0,"",
Z;WRT 4,A-2E5+A
;JMP 2+
50: GTO "X-INCRS"+
51: R1+C;R12+R1;C+R1
2;RCF Y+1+12R35,
R1,R12+
52: IF Y<11;GTO "Y-C
OUNT"+
53: RET +

```

SUBROUTINE LIST

```

66:      "LIST";1+Y;0+R+B
+0;FXD 0;35+X+
67:      X+1+X;0+RX;IF X<
47;JMP 0+
68:      Y+1+Y;0+X;FMT "@
0J00HSUBSTRATE N
0. 00I070",FXD
2,Z;WRT 4,Y-1+
69:      LDF Y,R1;LDF Y+1
2R35,R21+
70:      FMT "00J00J00J00
H      POSITION
      R(NOMINAL)
      R(T)      ",Z;
WRT 4+
71:      FMT "00H      (DELT
A-R)/R - (%)00J"
,Z;WRT 4+
72:      X+1+X;FXD 0;FMT
"00J00H      @
0I070",FXD 2,Z;
WRT 4,X+
73:      (R(X+20)-RX)/RX*
100+R(X+12)+
74:      FXD 4;FMT "00H
00I070",FXD 1
1,Z;WRT 4,RX+
75:      FMT "00H      00I07
0",FXD 11,Z;WRT
4,R(X+20)+
76:      FMT "00H      00I
070",FXD 11,Z;
WRT 4,R(X+12)+
77:      R(X+12)+R(X+35)+
R(X+35)+
78:      IF X<11;JMP -6+
79:      FMT "00J00J00J00
J",Z;WRT 4;FXD 0
;IF Y<12;JMP -11
+
80:      0+X;FMT "00H
      AVERAGE VALUES
      OF (DELTA-R)/R I
N %00J",Z;WRT 4;
FXD 2+
81:      X+1+X;R(X+35)/12
+R(X+35);FMT "00
J",8"00H ", "00I0
70",FXD 2.0,Z;
WRT 4,X+
82:      FMT "00H      00I
070",FXD 8.4,Z;
WRT 4,R(X+35)+
83:      IF X<11;JMP -2+
84:      RET +
85:      "STR";RCF 1,R33,
R35;END +
R60

```

DISTRIBUTION

	Copy
R. Bulcock, ERDA-KCAO	1
V. C. Vespe, ERDA-ALO	2
J. A. Freed, LASL	3
S. J. Buginas, LLL	4
H. M. Brinkmeier, Monsanto	5
J. T. Grissom, SLA	6
R. D. Hensley, D/131, FC39	7
A. K. Thoeni, D/131, FC39	8
J. D. Corey, D/554, BD50	9-10
L. Stratton, D/554, 2C44	11-13
R. F. Pippert, D/700, 1A42	14
R. P. Frohmberg, D/800, 2A39	15
J. S. Bosnak, D/842, MD40	16
J. A. Peters, D/842, MD40	17
L. E. Schantz, D/842, MD40	18-22
T. A. Wiley, D/842, MD40	23
L. W. Smith, D/844, BW31	24
R. L. Morgan, D/845, MF39	25
G. A. Yocum, D/862, MA40	26
R. E. Kessler, D/865, 2C40	27

BDX-613-1566

EP 04122656

LABORATORY AUTOMATIC DATA ACQUISITION SYSTEM,
L. E. Schantz, D/842, UNCLAS Final, September
1977.

A laboratory automatic data acquisition system (ADAS) has been established in the electronic products laboratory. The system consists of three basic blocks: the software control calculator, which controls the entire system, receives data from the system, and acts upon it in some intelligent manner; a variety of laboratory measuring equipment which gathers data and inputs it in BCD format; and the coupler/controller-and-patch-panel combination which makes it possible for BDC information

ELECTRICAL: Data Acquisition Systems

LABORATORY AUTOMATIC DATA ACQUISITION SYSTEM
L. E. Schantz, D/842, UNCLAS Final, EP 04122656,
BDX-613-1566, September 1977.

A laboratory automatic data acquisition system (ADAS) has been established in the electronic products laboratory. The system consists of three basic blocks: the software control calculator, which controls the entire system, receives data from the system, and acts upon it in some intelligent manner; a variety of laboratory measuring equipment which gathers data and inputs it in BCD format; and the coupler/controller-and-patch-panel combination which makes it possible for BDC information

LABORATORY AUTOMATIC DATA ACQUISITION SYSTEM,
L. E. Schantz, D/842, UNCLAS Final, EP 04122656,
BDX-613-1566, September 1977.

A laboratory automatic data acquisition system (ADAS) has been established in the electronic products laboratory. The system consists of three basic blocks: the software control calculator, which controls the entire system, receives data from the system, and acts upon it in some intelligent manner; a variety of laboratory measuring equipment which gathers data and inputs it in BCD format; and the coupler/controller-and-patch-panel combination which makes it possible for BDC information

from the gathering apparatus to be transmitted to the calculator. Approximately 50 software programs have been written for the system, including one which permits the ADAS system to scan resistors, measure their values, record the values, and permit the percent change in resistor values to be plotted. A program has also been written for data on temperature coefficients of resistance on thin film resistors.

from the gathering apparatus to be transmitted to the calculator. Approximately 50 software programs have been written for the system, including one which permits the ADAS system to scan resistors, measure their values, record the values, and permit the percent change in resistor values to be plotted. A program has also been written for data on temperature coefficients of resistance on thin film resistors.

from the gathering apparatus to be transmitted to the calculator. Approximately 50 software programs have been written for the system, including one which permits the ADAS system to scan resistors, measure their values, record the values, and permit the percent change in resistor values to be plotted. A program has also been written for data on temperature coefficients of resistance on thin film resistors.