



A data acquisition and recording system
by John Emil Somppi

A thesis submitted in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE
in Electrical Engineering
Montana State University
© Copyright by John Emil Somppi (1977)

Abstract:

When a program controlled data acquisition and recording system is desired, several important areas beyond a central processing unit, memory, and input/output must be considered. Specifically, these are: system control, interfacing of data, monitoring transducers and mass data storage. In this theses, designs for these additional features are presented. They are designed to complete one such data acquisition and recording system centered around the central processing unit, memory and input/output on an existing microcomputing system.

STATEMENT OF PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Montana State University, I agree that the Library shall make it freely available for inspection. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by my major professor, or, in his absence, by the Director of Libraries. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature

Date

John L. Sempke

May 20, 1977

A DATA ACQUISITION AND
RECORDING SYSTEM

by

JOHN EMIL SOMPPI

A thesis submitted in partial fulfillment
of the requirements for the degree

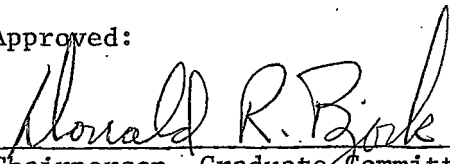
of


MASTER OF SCIENCE

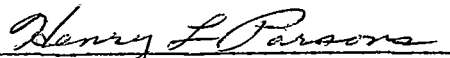
in

Electrical Engineering

Approved:


Chairperson, Graduate Committee


Head, Major Department


Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana.

May, 1977.

ACKNOWLEDGEMENT

In completing a project of this nature with its size and complexity, the talent and cooperation of a number of individuals all play a part in its success. Initially, I would like to thank the U.S. Forest Service, Forest and Range Experiment Station, Bozeman, Montana, and in particular Mr. William Hartsog for the financial funding of the project. Secondly, I would like to express my appreciation to Mr. Donald Bjork, project leader and chairperson of my graduate committee, for his assistance. Additionally, I wish to express my thanks to Mr. Darrell Irvin, Mr. Harry Plate, and the many others for their valuable technical assistance. Finally, I deeply appreciate the support I received from an untold number of friends. Their loyalty and support enabled the strength in the strenuous times. Although I might not have a chance to reciprocate, I know that it does not go unrewarded.

TABLE OF CONTENTS

<u>Chapter</u>		<u>Page</u>
1	INTRODUCTION.....	1
2	SYSTEM OPERATION.....	3
3	SYSTEM ARCHITECTURE.....	5
4	MICROCOMPUTER UNIT.....	8
	MICROCOMPUTER ARCHITECTURE.....	8
	MICROCOMPUTER OPERATION.....	8
	THE CENTRAL PROCESSING UNIT.....	9
	Instruction Format.....	9
	Timing and Control.....	10
	Bus Structure.....	11
	MEMORY SYSTEM.....	12
	INPUT/OUTPUT SYSTEM.....	13
	I/O Port Architecture.....	13
	I/O Port Operation.....	17
	REAL TIME CLOCK SYSTEM.....	18
	SUMMARY.....	18
5	FRONT PANEL UNIT.....	19
	FRONT PANEL OPERATION.....	19
	FRONT PANEL UNIT ARCHITECTURE.....	23
	FRONT PANEL KEYBOARD AND DISPLAY CIRCUIT.....	25
	DISPLAY MULTIPLEXER CIRCUIT.....	30

<u>Chapter</u>	<u>Page</u>
	FRONT PANEL CONTROL CIRCUIT..... 34
	FRONT PANEL DATA CIRCUIT..... 39
	Control Field 1 Instructions..... 43
	Control Field 2 Instructions..... 44
	Displaying Data Register Contents..... 45
	SUMMARY..... 46
6	DATA ACQUISITION SYSTEM..... 47
	ISOLATION SYSTEM..... 48
	DATA ACQUISITION INTERFACES..... 51
	Interface Module Addressing..... 52
	Eight Channel Analog To Digital Converter Module.... 55
	Binary Up/Down Counting Interface..... 58
	SUMMARY..... 62
7	MASS STORAGE SYSTEM..... 64
	MASS STORAGE SYSTEM ARCHITECTURE..... 64
	CASSETTE SYSTEM BUS..... 64
	CASSETTE TAPE DRIVE UNIT..... 66
	CASSETTE INTERFACE SYSTEM..... 69
	MASS STORAGE SYSTEM OPERATION..... 69
	Control..... 70
	Sense..... 70
	Reading And Writing..... 73

<u>Chapter</u>	<u>Page</u>
Control And Timing.....	74
INTERFACE CONTROL CIRCUIT.....	76
Power Up.....	81
Load.....	81
Stop.....	82
Fast-Wind.....	82
Read.....	83
Write.....	84
Gapping.....	85
Setting Incremental or Synchronous Modes.....	85
Gap Detection.....	86
Leader or Load Point Detection.....	86
Timing Generation.....	87
Interface Control Circuit Summary.....	88
INTERFACE DATA CIRCUIT.....	89
Interface Data Circuit Architecture.....	89
Synchronous Write Operation.....	91
Incremental Write Operation.....	94
Synchronous Read Operation.....	96
Incremental Read Operation.....	97
SUMMARY.....	98
8 CONCLUSIONS AND RECOMMENDATIONS.....	99

<u>Appendix</u>	<u>Page</u>
I LOGIC CONVENTIONS.....	111
II FRONT PANEL MINIPROGRAMS.....	113
BIBLIOGRAPHY.....	120

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	CPU CONTROL SIGNAL DESCRIPTION.....	11
2	CONTROL FIELD 1 FUNCTIONS.....	42
3	DAS BUS CONFIGURATION.....	52
4	CONTROL INPUT LINES.....	67
5	STATUS LINES.....	68
6	INPUT/OUTPUT LINES.....	68
7	TAPE UNIT CONTROL OPERATIONS.....	71
8	STOP/RUN-MODE OPERATIONAL STATUS STATES.....	72
9	TAPE UNIT OPERATIONAL CONTROL SIGNAL STATES.....	77

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1	DAS SYSTEM ARCHITECTURE.....	6
2	CONTROL BYTE FORMAT.....	14
3	OUTPUT I/O FLAG HANDSHAKE.....	15
4	INPUT I/O FLAG HANDSHAKE.....	16
5	STATUS BYTE FORMAT.....	16
6	FRONT PANEL KEYBOARD AND DISPLAY SCHEMATIC.....	26
7	FRONT PANEL DISPLAY MULTIPLEXER SCHEMATIC.....	31
8	FRONT PANEL CONTROL SCHEMATIC.....	35
9	FRONT PANEL DATA SCHEMATIC.....	40
10	MICROINSTRUCTION FORMAT.....	42
11	ISOLATION SYSTEM SCHEMATIC.....	50
12	INPUT FLAG DELAY TIMING DIAGRAM.....	51
13	DAS INTERFACE MODULE ADDRESSING CIRCUIT.....	54
14	EIGHT CHANNEL ANALOG TO DIGITAL CONVERTER SCHEMATIC.....	57
15	BINARY UP/DOWN COUNTER SCHEMATIC.....	61
16	CASSETTE INTERFACE BUS.....	65
17	CASSETTE STATUS BYTE FORMAT.....	72
18	SYNCHRONOUS GAP-DATA FORMAT.....	74
19	INCREMENTAL GAP-DATA FORMAT.....	76
20	INTERFACE CONTROL SCHEMATIC.....	80
21	SYNCHRONOUS WRITE TIMING DIAGRAM.....	91

<u>Figure</u>		<u>Page</u>
22	INTERFACE DATA CIRCUIT SCHEMATIC.....	92
23	INCREMENTAL WRITE TIMING DIAGRAM.....	95
24	SYNCHRONOUS READ TIMING DIAGRAM.....	96
25	INCREMENTAL READ TIMING DIAGRAM.....	98

ABSTRACT

When a program controlled data acquisition and recording system is desired, several important areas beyond a central processing unit, memory, and input/output must be considered. Specifically, these are: system control, interfacing of data, monitoring transducers and mass data storage. In this theses, designs for these additional features are presented. They are designed to complete one such data acquisition and recording system centered around the central processing unit, memory and input/output on an existing microcomputing system.

CHAPTER 1

INTRODUCTION

In the field of real time data acquisition, there are systems which cannot be monitored by conventional means. Either they are too complex or too dynamic for data to be adequately compiled by human observation. A logical alternative then is to automate the data acquisition process.

To fulfill this problem the Electronics Research Laboratory of Montana State University was approached by the United States Forest Service and asked to build an automatic data acquisition system to fit the following specifications:

1. The system must be able to monitor and store data while operating with little human interaction.
2. The system must have some intelligent decision making capability for real time data evaluation.
3. It should be able to monitor moderately high speed systems. (Data rates up to 3 K Hz.)
4. It should be versatile enough to effectively monitor various data acquisition situations.
5. It should be small in size to allow ease in portability.
6. Power consumption should be low, as it would be battery powered
7. It should be rugged enough to withstand an outdoor operating environment.

To meet these criteria, we engineered and built a microcomputer based data acquisition system. Its computer-like operation and architecture made it a natural to fit the operational and system specifications. The electrical and mechanical design of the system using low power electronics and small, sturdy, modular construction fulfilled the criterion.

CHAPTER 2

SYSTEM OPERATION

In the data acquisition process it was desired to have the system do all the data monitoring and data accumulation with a minimum of human intervention. As a result, the operation of the data acquisition system centers around the programmable microcomputer. It is this microcomputer which monitors and processes the data from its various sensors (or transducers). It retains as accumulated data pertinent information it has monitored. This information can later be retrieved for detailed evaluation. This process is controlled by a system program executed by the microcomputer. This operation of the microcomputer in turn is under complete operator control.

In a typical situation, the operator initiates microcomputer operation by loading the operating program into the microcomputer from the mass storage medium. This storage medium is the permanent storage area where operating programs as well as retained monitored data is kept for permanent continual or future referral. With the execution of this controlling program, the operation of the data acquisition begins. The operator may interact with the system either in supplying necessary parameters for operation or by examining monitored data if desired. However, the data acquisition is primarily controlled by the operating program.

As data is being accumulated, minor analysis may be done, but primarily pertinent data is being accumulated in the mass storage medium.

Here it remains for later referral. When detailed examination and analysis of this data is desired it is transferred from the mass storage area to larger computing facilities for evaluation.

The primary task for this system is data acquisition. In-depth analysis of the data is left to larger computing systems more specifically suited to that task.

CHAPTER 3

SYSTEM ARCHITECTURE

The architecture of the Data Acquisition and Recording System (DARS) is modular in concept. Fundamental functions in the operation of DARS are done by separate units. These units then interact to perform the complete data acquisition process. With the general system operation distributed between these subsystems, no particular portion of the system is overtaxed and a high degree of versatility is maintained.

A diagram of DARS is shown in Figure 1. There are four subsystems comprising the system. The primary one of these is the microcomputer system. It is the intellect of the system. It contains and executes the controlling program governing DARS operation. The microcomputer system is also the decision making portion of the system. It is the heart of DARS as it enables and directs the operation of the monitoring and data accumulating operations.

Working closely with the microcomputer system is the Front Panel Unit (FPU), which is used for system control. This is the system which controls the microcomputer system operation. It is the primary interface between the operator and the data acquisition system. System operation can be initiated, terminated, monitored or modified by operator action with this unit.

The Data Acquisition System (DAS) is the portion of the DARS through which all monitoring of data is done. It is the interface between the data being monitored and the microcomputer. The purpose of

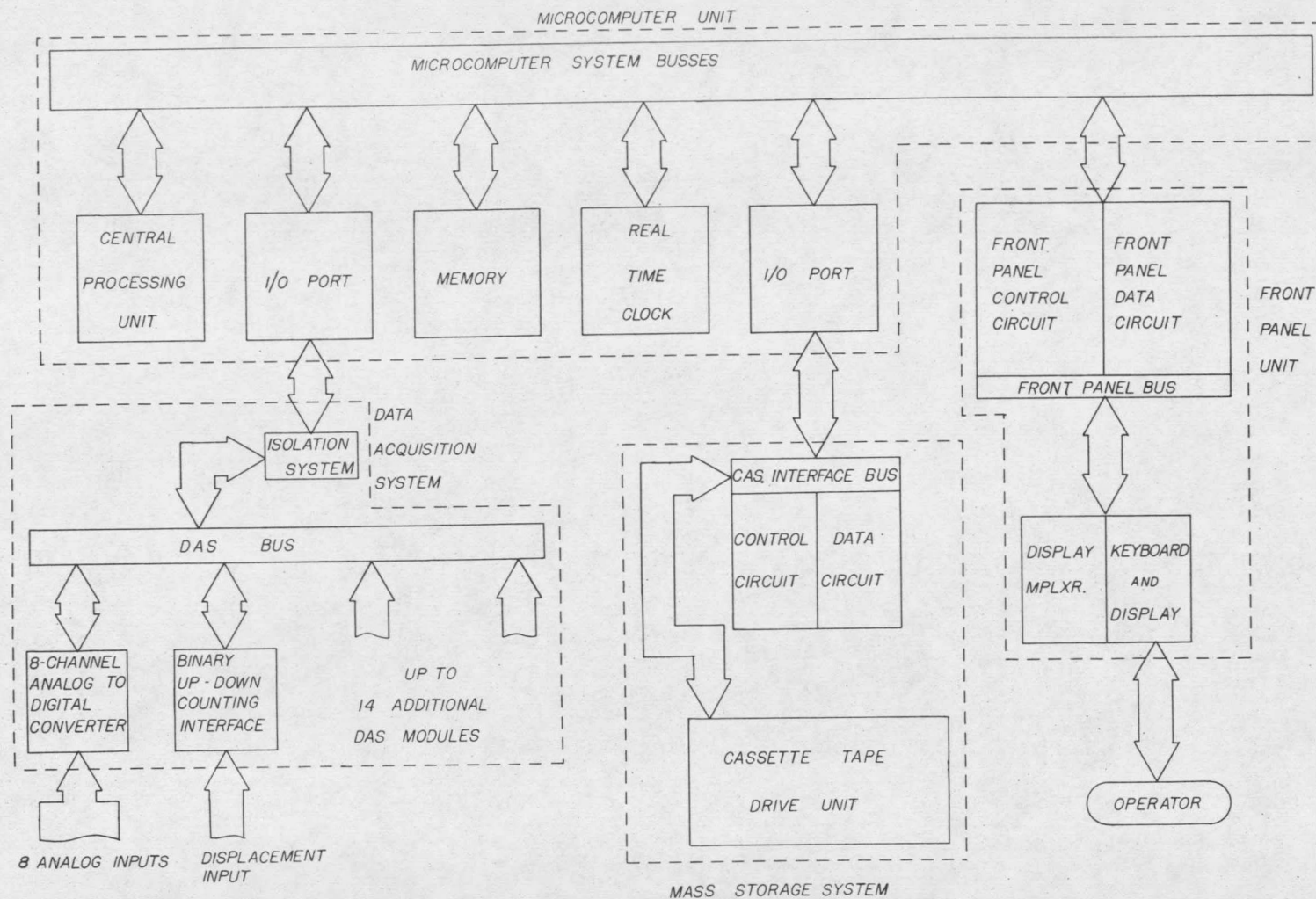


FIGURE 1: DARS SYSTEM ARCHITECTURE

of this system is to change the form of the monitored data into a meaningful representation understood by the microcomputer.

The final subsystem of DARS is the mass storage medium. This system is used for semipermanent information storage. Operating under microcomputer control information may either be stored on or retrieved from this unit. This information may be the data monitored by DARS being retained for later use or an actual program used to control the microcomputer system. Using this type of system the entire operation of DARS becomes independent and self-contained.

CHAPTER 4

MICROCOMPUTER UNIT

The microcomputer unit is responsible for complete operation of the system. It is a basic low-power microcomputer with processor, memory, and input/output capability. This system was designed by Darrell B. Irvin (1).

MICROCOMPUTER ARCHITECTURE

The architecture of the microcomputer is a bus structured system consisting of four major sections. These sections are the Central Processing Unit (CPU), Memory, Input/Output ports (I/O) and a Real Time Clock (RTC). These are all connected by the system busses. These busses are the Data Bus (D Bus), the Address Bus (HL Bus) and a Control Bus.

MICROCOMPUTER OPERATION

The Central Processing Unit controls the operation of the Microcomputer system. It performs all the arithmetic and logical operations as well as controls program execution when in operation. It interacts with the three other supporting microcomputer sections through the bus systems. The CPU fetches program instructions as well as stores and retrieves data from the Memory section of the Microcomputer. It is able to interact with devices external to the Microcomputer system, such as the Mass Storage Unit or the Data Acquisition System, through the Input/Output ports. I/O is the link between the CPU and the external systems.

The Real Time Clock aids the system by enabling the monitoring of time as well as data. Measuring the passage of time is necessary in acquisition, but it is extremely difficult to do accurately with only the CPU. The RTC accurately measures time under control of the CPU.

THE CENTRAL PROCESSING UNIT

The power of the CPU lies with an Intel 8008 Microprocessor. The 8008 is an eight bit parallel microprocessor. It can address up to 16 kilobytes (16K bytes, one K byte is 1024 bytes) of memory and has complete I/O capability. It may operate with external interrupts using a vectored interrupt scheme. It is a moderate speed microprocessor well suited to moderately high speed data acquisition.

Instruction Format

The processor instructions are in the multibyte format. Instruction size varies from one to three bytes depending on the instruction. One memory cycle is used to fetch each byte from memory. The time of each cycle also varies depending upon the complexity of the instruction being executed.

Single cycle instructions include non-memory reference operations. These instructions need no additional data other than the instruction for execution. Two and three cycle instructions are memory reference instructions and I/O instructions. The memory reference instructions are two cycles if an 8 bit data byte is needed. They might be three cycles if a 16 bit two byte address is required. One 8 bit byte is

retrieved each memory cycle. The first cycle is referred to as the Instruction Cycle since the instruction is fetched during this time. The second and third cycles are termed Data Cycles since data is fetched at this time. All I/O instructions are 2 cycle with data input from or output to the I/O port in the second cycle or the I/O Cycle.

Timing and Control

Any instruction contains from 3 to 5 separate state times per memory cycle. The first two states are always the outputting of the low and high order bits respectively of the program counter from the micro-processor to memory. The third state is the fetch phase, where the contents of memory at the address indicated by the program counter are sent to the CPU. The fourth and fifth times are optional, depending upon the instruction being executed. The instruction is executed following the fetching of the memory information of the final memory or I/O cycle. These five states' times are referred to as T1, T2, T3, T4, and T5 respectively.

In addition to these five states, there are three more states: T1I, STOP and WAIT. T1I has the same function as T1 except it indicates that the processor is responding to an interrupt request. STOP indicates that the CPU is not executing any instructions; it has been halted. WAIT indicates the processor is waiting between T2 and T3 for a slow device or memory to furnish the data being fetched.

Besides these eight timing signals, there are other control signals

generated or monitored by the CPU. These control signals which are of interest to DARS operation are shown in Table 1 with their functions. The Control bus contains all state time and CPU control signals.

TABLE 1

CPU CONTROL SIGNAL DESCRIPTION

<u>CPU CONTROL SIGNAL</u>	<u>FUNCTION</u>
CLR	Initializes and resets all bus connected units
I/O	Indicates and input or output processor cycle
DATA	Indicates a data fetch from memory processor cycle
INST	Indicates an instruction fetch from memory processor cycle
DSTB	Portion of any timing state when bus data is transferred
DRST	Signal to reset data transmission circuitry following DSTB in T3
INT	Interrupt request to the CPU
PRIOUT	Priority Out signal enabling an interrupt request may be made if necessary
PRST	Priority Reset signal enabling PRIOUT
<u>DMA</u>	Acts to disconnect the memory address register on the CPU from the H-L Bus

Bus Structure

The 8008 has one 8 bit parallel data port through which the Data and Address Busses are multiplexed. It is a bidirectional port through

which all addresses, instructions and data pass.

The Address Bus or HL Bus is a 14 bit unidirectional bus. It is comprised of two smaller busses, the H Bus and the L Bus, six bits and eight bits wide respectively. The H Bus contains the six high order address bits output to memory during a fetch cycle. The L Bus contains the eight lower order address bits during a fetch cycle. Also, during an I/O cycle data output from the processor is output on the L Bus. By concatenating the H and L Busses the full 14 bit address to memory is formed.

The Data Bus or the D Bus is an eight bit bidirectional bus to the processor. All data input to the processor is done on the D Bus. This data may be either instructions or data bytes from Memory, I/O Ports or the RTC.

In the case of the D and HL Busses, the data sense on each bus is positive. (For further explanation of level and data senses see Appendix I). When the D and HL Busses are not in use, however, they will be at a true or high level. When valid data is on any bus it will be in the upright form or positive true sense.

MEMORY SYSTEM

The memory system for DARS consists of a maximum of 16K addressable bytes of data. The upper 2K bytes, address locations of 3800' through 3FFF', (the suffix " ' " denotes base 16 or hexadecimal notation), are Read Only Memory (ROM). Here special tables and permanent programs may

be stored. The remaining lower 14 bytes is Random Access Memory (RAM). This area is used for temporary data and program storage.

INPUT/OUTPUT SYSTEM

The scheme used in the microprocessor unit for data transfer to and from the CPU centers around the Input/Output Port. The I/O Port is designed for bidirectional data transfer between the microcomputer processor and the I/O device with additional device control and operation monitoring capability. This enables the processor not only to exercise data interaction with the I/O device but also control and monitor the device's operations. The I/O Port is not limited to interacting with a single device however, as it will be shown in the Data Acquisition System that multiple input and output devices can be serviced through a single port.

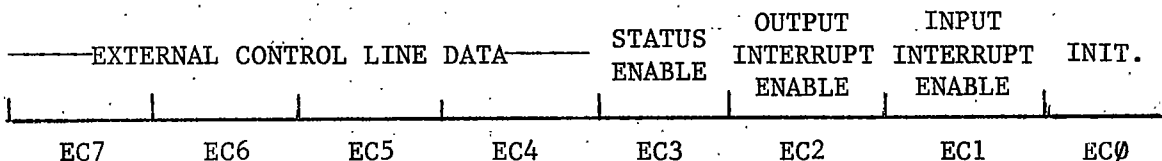
The input, output, control and status monitoring sections of an I/O Port are completely separate. Each function may be exercised separately without affecting the other functions. Inputting and outputting of data is done on a single byte basis. Data is transferred a byte at a time. Due to the separate nature of input and output portions, however, byte outputting may be done simultaneously. Control and status monitoring is done by External Control and External Sense lines respectively between the port and the device.

I/O Port Architecture

The input, output and control sections of the I/O Port have

individual unique addresses which are different from any other port section. In this manner each can be accessed independently by the processor. The architecture of the I/O Port reflects this. These are the major sections of an I/O Port, the input section, output section and control section.

The control section directs operation of both the I/O port and the I/O device. Its center is a control data register which contains a data byte output by the processor to the port. One half (or four bits) of this control byte is used for controlling port operation while the other half is used for the External Control line information. These External Control lines direct the operation of the I/O device. Figure 2 shows the format of the control data byte.

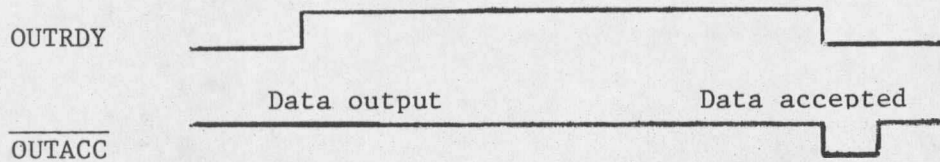


CONTROL BYTE FORMAT

FIGURE 2

In a similar manner, the output section of an I/O port utilizes a output data register. This register contains the data byte output from the processor. Here the data byte is stored for access by the I/O device. Two additional communication lines are used in the output section to facilitate the data transfer. One is the Output Ready line (OUTRDY). OUTRDY indicates to the device that data output from the processor is

ready to be transferred. The second line indicates to the port that the output data has been accepted by the output device. It is called the Output Accepted line ($\overline{\text{OUTACC}}$). (The notted signal name indicates the signal has inverted logic sense. See Appendix I for further explanation.) A handshake process between the I/O port and the I/O device using OUTRDY and $\overline{\text{OUTACC}}$ enable a proper data transfer. Figure 3 shows the state of these flag lines during an output process.

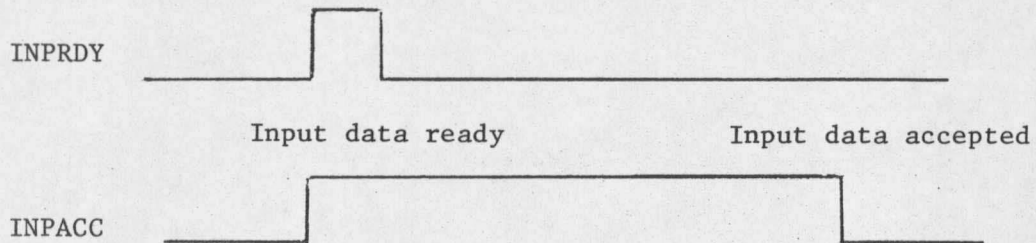


OUTPUT I/O FLAG HANDSHAKE

FIGURE 3

The architecture of the data input section of the I/O port is similar to the output section. It utilizes an input data register and two flag lines, Input Ready (INPRDY) and Input Accepted (INPACC). The input data register is used to store the input data byte from the inputting device until it is accepted by the processor. When data is available from the inputting device, INPRDY indicates such. The data byte is stored then in the input data register. When the processor inputs the data from the I/O port register, INPACC reflects so. A handshake using INPRDY and INPACC complete the data transfer. Figure 4 details the state

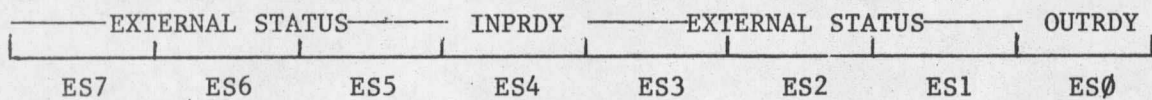
of these flag lines during an input cycle.



INPUT I/O FLAG HANDSHAKE

FIGURE 4

The monitoring of device status is done through the input section of the I/O port and controlled by the control section. The setting of the status enable bit of the control byte indicates that status data will be input during the next processor input request. In place of the contents of the input data register, data on the six External Sense lines along with the states of OUTACC and INPRDY are relayed to the processor. In this manner, I/O device status as well as the input and the output handshake conditions can be monitored. Figure 5 shows the format of the status byte relayed to the processor when I/O port status is requested.



STATUS BYTE FORMAT

FIGURE 5

A fourth section of the I/O port is the interrupt section. It is used in input and output operations to facilitate the handshake process. Either operation may be completed using processor interrupts to gain the attention of the processor when a handshake is complete. Interrupt operation of the I/O port is not necessary in all situations and is controlled by the state of the control data register. Output interrupts and input interrupts are separately selectable by this manner.

I/O Port Operation

Initially in either input or output operations, the I/O device must be placed into the proper operational state. This is done by outputting the appropriate control data to the control data register of the I/O port. The operational state may be monitored by the status information supplied through the port. In an output operation, data is output to the port from the processor, stored in the output data register and sets OUTRDY. This indicates to the output device that data is available. When the device accepts data from the port it sets OUTACC. This indicates to the processor by either an interrupt or through monitored status that the output transfer has been completed and another one may be initiated. In a like manner, during an input operation, the I/O device indicates that it has data available by setting INPRDY. This stores the input data in the input data register of the port. It also indicates to the processor by interrupt or monitored status line the availability of input data. Upon the inputting of this data by the

processor INPACC is set, completing the input transfer, enabling another transfer to be initiated. These byte-by-byte data transfers continue during the I/O process. When the process is complete, the operation of the I/O device might be modified by changing the control data appropriately in the control data register.

REAL TIME CLOCK SYSTEM

The RTC is a programmably controlled interval timer. Under processor control, various time intervals may be initiated to accurately measure time. These intervals span from one millisecond to 2550 seconds. At the completion of the specified time period, a processor interrupt is generated. These intervals may either occur once or be repeated one after another, increasing the flexibility of the time keeping aspect of DARS. In addition to this feature, the RTC has an accurate crystal oscillator time base for a self-contained accurate frequency standard available to other units in DARS.

SUMMARY

The Microcomputer Unit is the center of DARS. Its computer-like architecture lends itself to high versatility and ability. Its Central Processing Unit gives the system intelligence capable of real time decision making and data manipulation. The system's data input and output capability coupled with its programmable operation enable it to fit into numerous data acquisition situations with minimum modifications.

CHAPTER 5

FRONT PANEL UNIT

In the operation of a programmed processing unit, there must be some device through which it is possible for an operator to exert external control over the processor's operation. This control could vary from program or process stopping and starting to the examining and/or modifying of internal storage areas. Thus a front panel was incorporated in the system.

FRONT PANEL OPERATION

All operator interaction with DARS is done through the Front Panel Unit. The FPU contains both keys through which data and control commands may be entered and displays which may furnish various types of system information. With the keys for input and the displays for output, the front panel is the primary I/O device between the system and the operator.

Typical FPU operations are initiating or halting processor activity, modifying or examining processor data registers and modifying or examining memory data. In doing this, the FPU controls the operation of the processor, using it to manipulate data, registers, memory, etc. It accepts data or furnishes data to the processor in completing the desired task. In this manner, data manipulation and processor control may be done through the FPU.

The FPU may be operated by a second means other than from the keyboard. FPU operations may be initiated through special commands by the

CPU. This enables programmable control of the FPU to accomplish tasks such as displaying or reading data from the keyboard. So in addition to being a tool used by the operator to affect the microcomputer, the FPU may be used by the microcomputer to supply the operator with system information.

When controlling processor operation, the FPU acts to source instructions to the processor, which are designed to accomplish the desired front panel function. A group of these instructions which accomplish the desired FPU operation is called a miniprogram. In addition to this, the FPU supplies or stores data during the Read, Write or I/O cycles of these instructions. In this manner, data necessary for the operation is either supplied by or stored into the registers in the FPU. This is how the Front Panel Unit examines or modifies data or operation of the microcomputer system. The miniprograms are stored in the Front Panel Unit. They contain both processor instructions and instructions to control the manner in which the FPU responds to activity of the processor.

All operator action is initiated by keystrokes on one or more of the 31 keyboard keys located on the DARS front panel. Sixteen of the keys are hexadecimal formatted numerical entry keys. These are used when entering numerical data into a front panel data register, called the Switch Register. This data register serves as the location for all operator initiated input data.

When system control is initiated from the front panel, it is done by

one of two ways. A two keystroke sequence may be done or a special function key may be struck. In the two keystroke sequence, a prefix key is struck, followed by striking a numerical key. There are two function prefix keys, labeled I and II. When a keystroke on either of the two prefix keys precedes a keystroke on a numerical entry key, an FPU function is executed, rather than a data entry into the Switch Register. With this keying sequence, up to 32 separate functions may be initiated by the operator.

Ten of these 32 front panel operations may also be initiated by striking a function key. These ten function keys parallel ten of the 32 function keystroke sequences, and are the more commonly used FPU functions. These are: the initiation or termination of program execution (RUN, HALT), loading or displaying of the processor's program counter (LPC, DPC), loading or displaying the accumulator (LDA, DA), loading or displaying the H and L registers (LHL, DHL), and loading or displaying the contents of the next memory location following the location indicated by the address in the H and L registers (LNM, DNM). These two function keys are used to decrease the complexity of executing these more commonly used FPU functions.

The three final keys on the keyboard are display cycle (DX), display clear (CD), and system clear (CLR). DX changes the data shown on the display. CD clears the contents of the Switch Register to zeroes and it cancels a keystroke on a function prefix key. CLR initializes or resets

the DARS system. There is a keyboard operated service switch on the front panel which can lock out the CLR button from clearing the system when depressed. CLR will only operate when enabled through the service switch.

In supplying data or system status information, DARS uses the front panel displays. For numerical data, the numerical display is used. To indicate system status, the front panel uses status indicator lamps. The numerical display is a four digit hexadecimal formatted array. It is able to display one of three separate types of data. This data is stored in three front panel system registers. These three registers, the Switch Register, the Address Register and the Data or D Register, are used for front panel numerical data. Each time DX is struck, the display rotates through displaying one of these three registers and a blank state, one state at a time. The Switch Register indicates data keyed in through numerical entry from the keyboard. This register is four hexadecimal digits in length. The Address Register is also four digits in length and is used to display the 14 bit processor address data. The Display Register is two digits in length and is used to display single bytes of information. These registers are used to display data obtained by execution of various FPU functions. The blank state is used when the display is not in use to conserve power. CD or a numerical entry keystroke automatically changes the state of the display to indicate the contents of the Switch Register. Executing the RUN function or the transition of

CPU activity from being halted to executing places the display into its blank state.

To indicate system status, the front panel uses three groups of status indicator lamps. These are the Cycle, Flag and System Status indicators. The Cycle indicator lamps are labeled READ, WRITE, INST and I/O. These display the current processor memory cycle state. The Flag indicator lamps are C, Z, S and P. These indicate the processor flag states at the time of the last FPU initiated processor halt or flag polling. As with the CLR key, the Cycle and Flag indicators are only enabled by the service switch. This is done to conserve power, as these indications are not always necessary in DARS operation. The System Status indicator lamps are PWR, MSG, KEYBD, RUN, TAPE and I,II. PWR indicates system power is on. MSG directs the operator that appropriate action, as per display information, should be taken. KEYBD indicates the four digit display currently indicates the Switch Register data. RUN shows the CPU is operating. TAPE indicates the mass data storage unit is either without tape or the unit door is open. I,II is the function prefix indicator, displaying that a function prefix key has been struck. These System Status indicators are always operational when the DARS system is operating.

FRONT PANEL UNIT ARCHITECTURE

There are two major systems in the architecture of the Front Panel Unit. The first is the Keyboard/Display System. This is the primary system used to initiate FPU operations, supply data for FPU operations.

and to display pertinent microcomputer information. It is oriented toward interaction with the DARS operator. It has a keyboard from which numerical data may be entered or up to 32 separate FPU operations may be initiated. Its display portion is used to supply information from the data storage registers in the FPU. System information is supplied to the operator in this manner.

The second system is the Control/Data System. This portion of the FPU interacts with the Keyboard/Display System and the microcomputer system. It also contains data storage registers which store or supply processor data during FPU operations. Data from the keyboard and processor is stored here. These are among the registers which accessed by the front panel displays for operator information. Action by the Control/Data System may be initiated by the Keyboard/Display System for operator controlled tasks or by the CPU for processor controlled tasks.

When operating in conjunction with the Control/Data System, the Keyboard/Display System continually updates the Switch Register and initiates Control/Data System action when a FPU function has been keyed. Also display information from the Control/Data System is reflected by the front panel displays.

The Front Panel Unit is a bus connected device. All interaction between it and the microcomputer system is done via the bus system. As a bus connected device, it is able then to not only access the CPU, but all other bus connected systems. This enables the FPU to interact with

memory or even the I/O ports if necessary. The FPU also has an internal bus, the Front Panel Bus, or mini bus. It connects the two sections of Control/Data System with themselves and to the Keyboard/Data System. The FPU bus is a series of smaller busses and control signal lines used for data and control purposes by the front panel systems.

Both front panel systems have two subsystems. The Keyboard/Display System is made up from the Keyboard and Display circuit and the Display Multiplexer circuit. The Front Panel Control circuit and the Front Panel Data circuit comprise the Control/Data System. These four circuits are described in the following sections. This shall more completely describe front panel operation.

FRONT PANEL KEYBOARD AND DISPLAY CIRCUIT

The Front Panel Keyboard and Display circuit, shown in Figure 6, contains displays, display drivers, keys and keystroke detection circuitry. Its primary functions are to indicate to the Display Multiplexer circuit the occurrence of numerical or functional keystrokes and to illuminate front panel displays under the control of the Display Multiplexer.

When detecting a depressed key, the Front Panel Keyboard and Display circuit uses four control signals and a five bit Keyboard Bus to initiate Display Multiplexer action. The control signals are VALID, Function Flag (FFLG), BIT FLAG and EQUAL. The Keyboard Bus indicates the numerical value of the depressed key for numerical entry or which of the 32 different front panel functions is being initiated. A true

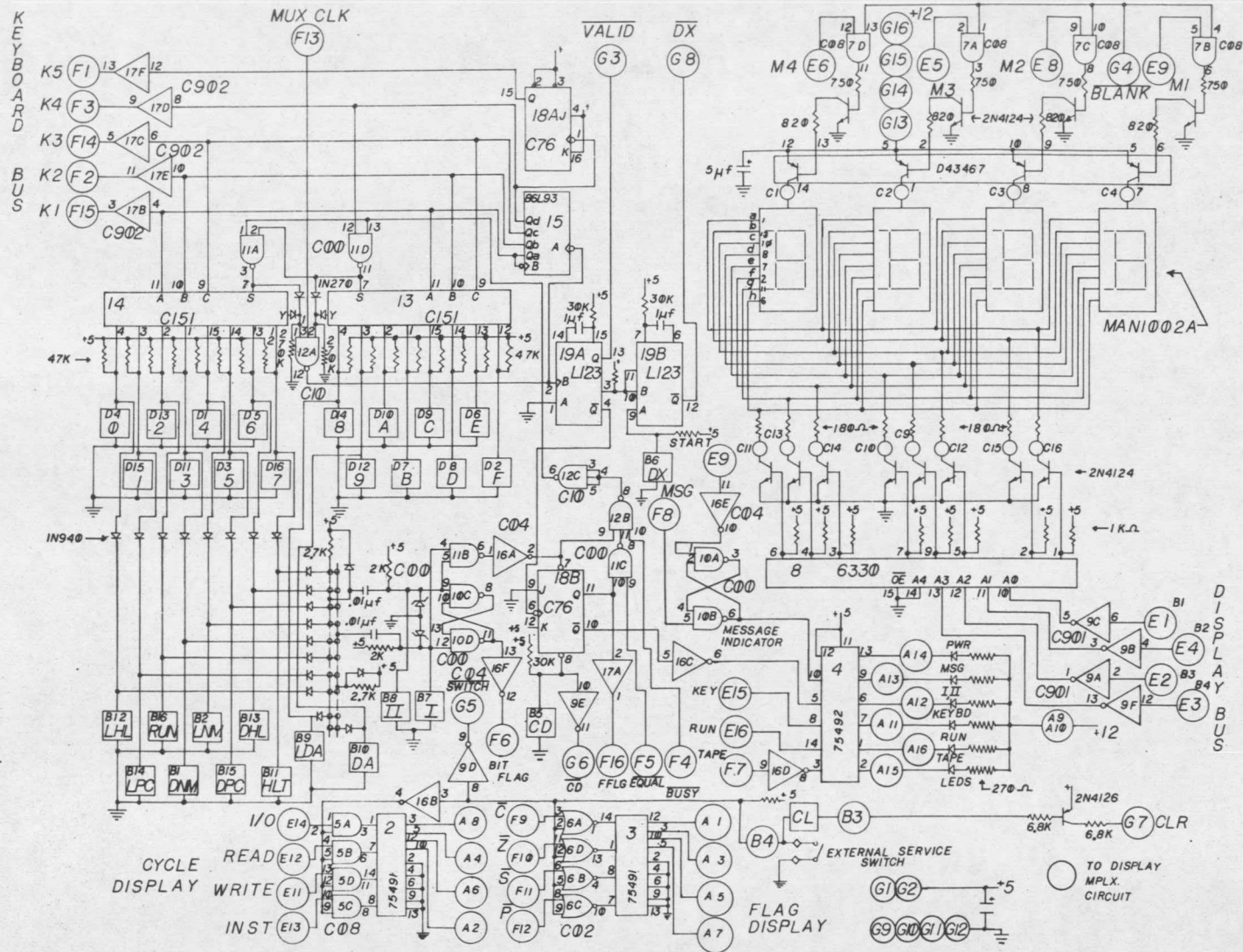


FIGURE 6: FRONT PANEL KEYBOARD AND DISPLAY SCHEMATIC

state of VALID indicates a key is depressed and the Keyboard Bus contains the correct data. FFLG being true distinguishes a front panel function keying sequence from a numerical entry. The state of BIT FLAG indicates which prefix key has been struck. If BIT FLAG is false, the I prefix was struck last. A true BIT FLAG indicates the prefix II. EQUAL indicates the sense of BIT FLAG is the same as the fifth bit (K5) of the Keyboard Bus.

Numerical keys are continually scanned by the keystroke detection circuit, a single key at a time. The lower four bits of the Keyboard Bus indicate the numerical value of the key being scanned at that moment. In a numerical entry, when a depressed key is detected, further scanning is inhibited and VALID becomes true. Ignoring the fifth Keyboard Bus bit, the lower four bits of the bus contain the numerical value of the depressed key. When the key is released, VALID becomes false and scanning resumes.

In detecting a front panel function keying sequence, the prefix key sets FFLG when struck and appropriately sets the sense of BIT FLAG. When the striking of a numerical key follows a prefix, VALID becomes true when the lower four bits of the Keyboard Bus indicate the value of the numerical key and the fifth bit is the same as BIT FLAG. Keyboard scanning is stopped until the numerical key is released. The releasing of the key causes VALID and FFLG to become false and normal keyboard scanning for numerical entry is resumed.

The operation of initiating a front panel function by striking a function key parallels the two keystroke method. When one of the function keys is depressed, a prefix pulse appropriately sets the sense of BIT FLAG and sets FFLG. Also, a numerical key sense line is activated, enabling the appropriate numerical value to be detected when the keyboard is scanned. $\overline{\text{VALID}}$ becomes true when the Keyboard Bus reflects BIT FLAG and the numerical value. Releasing the function key terminates $\overline{\text{VALID}}$, and resets FFLG and resumes normal keyboard scanning.

The three remaining keys do not initiate front panel functions, but control the state of the display, the Switch Register, or the DARS system. Depressing DX generates a pulse on the control line $\overline{\text{DX}}$. This directs the Display Multiplexer circuit to change the display state. CD activates the $\overline{\text{CD}}$ control line causing the zeroing of the data in the Switch Register. Similarly, CL activates the clear line on the system bus (CLR), and initializes the entire microcomputer system.

In illuminating the front panel displays, the Front Panel Keyboard and Display circuit drives both the numerical and status indicator displays. The Cycle status indicators are driven by the DARS system memory cycle control lines $\overline{\text{READ}}$, $\overline{\text{INST}}$, $\overline{\text{WRITE}}$ and $\overline{\text{I/O}}$. These are DARS Bus signals. Every memory cycle is reflected by the status displays. The Flag displays are driven by the four control lines $\overline{\text{C}}$, $\overline{\text{Z}}$, $\overline{\text{S}}$ and $\overline{\text{P}}$, generated by the FPU Data circuit. The key operated service switch enables the Flag and Cycle status indicators. Unless this external service switch is

closed, the Flag and Cycle status indicators will not operate. Of the System Status indicators, KEYBD and RUN are driven from signals used in the Display Multiplexer circuit. TAPE is generated by the mass storage circuitry. MSG and I,II are generated in the Front Panel Keyboard and Display circuit. MSG indicates the state of the Message Indicator flip-flop. This flip-flop is set by the $\overline{\text{MSG}}$ control line from the Front Panel Data circuit. Executing the front panel operation to turn on the MSG indicator sets this flip-flop, turning on the indicator. The indicator is turned off by the START control line. This resets the flip-flop when the processor resumes operation after halting.

The numerical displays are controlled by the Display Multiplexer circuit. These displays are operated in a digit multiplexed fashion, with the digits illuminated one at a time. The Display Bus contains the binary value of the digit currently being displayed. There are four multiplex phase (MUX PHASE) lines controlling the four multiplexed digits. These lines (M1, M2, M3 and M4) each enable a digit of the display to turn on and display the alphanumeric representation of the value contained on the Display Bus. The binary value on the Display Bus is decoded to appropriately light the display segments and represent the number. With the MUX PHASE lines enabling the displays and the decoded Display Bus value enabling display segments, the numerical display properly represents data. To blank the displays, the MUX PHASE lines are overridden by the $\overline{\text{BLANK}}$ signal. It disables all numerical displays from turning on, causing a blank display.

DISPLAY MULTIPLEXER CIRCUIT

The primary purpose of the Display Multiplexer circuit, shown in Figure 7, is to control Display Bus activity, thereby controlling numerical display illumination. In addition to this, it also initiates proper front panel action upon the recognition of valid keystrokes and control Switch Register data.

When a valid keystroke has occurred, it is indicated by $\overline{\text{VALID}}$ being true. When this occurs, the Display Multiplexer circuit can take one of two possible courses of action. If it was a numerical keystroke to place data into the Switch Register, FFLG is false and the Display Multiplexer circuit stores the four least significant bits of the Keyboard Bus into the Keyboard Display Register. This is a four digit serial shift register which contains the numerically keyed switch data. It is the contents of this register that are displayed on the numerical display when displaying Switch Register data. The contents of the Keyboard Display Register continually update the Switch Register, contained in the Front Panel circuit. This insures identical data in both registers at all times.

Secondly, in a keying sequence of a FPU function, the Display Multiplexer circuit initiates proper action to be taken by the Control/Data System. In such a key sequence, FFLG is true. The Display Multiplexer circuit generates the $\overline{\text{EQUAL}}$ signal, indicating bit K5 of the Keyboard Bus and BIT FLAG are the same. When $\overline{\text{VALID}}$ becomes true the control sig-

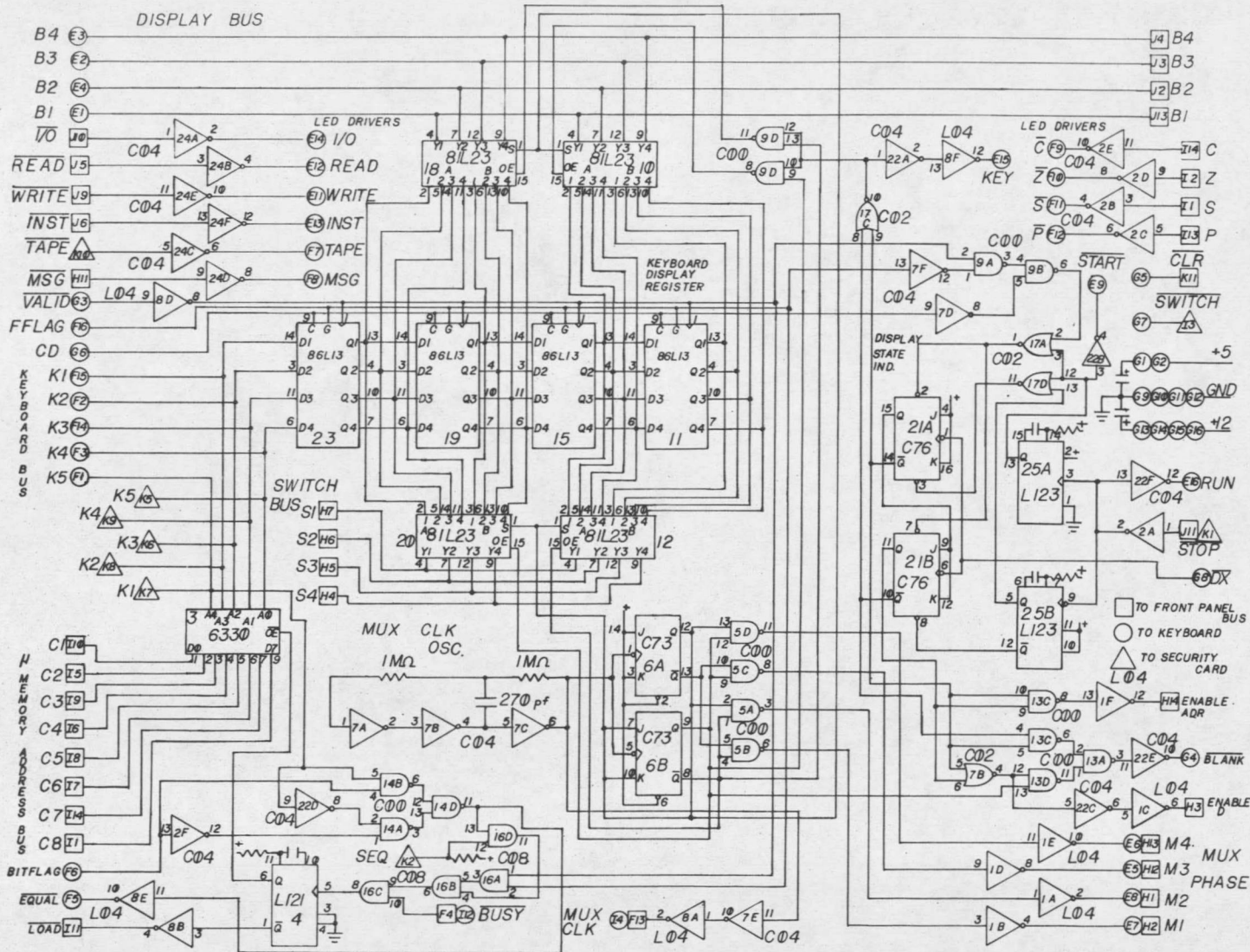


FIGURE 7: FRONT PANEL DISPLAY MULTIPLEXER SCHEMATIC

nals to the Control/Data System are generated. These control signals are an 8 bit code and a pulse on the $\overline{\text{LOAD}}$ signal line. The 8 bit code is an address used by the Control/Data System to ascertain which front panel function is to be executed. It is sent to the Control/Data System via the Micromemory Address Bus. The pulse on the $\overline{\text{LOAD}}$ line indicates that the Micromemory Address Bus contains valid information and to initiate the front panel function.

Two signals may inhibit $\overline{\text{LOAD}}$ and the 8 bit address from initiating Control/Data System action. These are $\overline{\text{BUSY}}$ and $\overline{\text{SEQ}}$. $\overline{\text{BUSY}}$ indicates that the FPU is currently executing a front panel function. In this case, $\overline{\text{LOAD}}$ is delayed until the current function is completed. $\overline{\text{SEQ}}$ is currently unused, but is provisional for future expansion of the FPU system to include a security system. This feature could be used to inhibit certain functions unless the service switch is closed.

Switch data is stored in the Keyboard Display Register. This data is continually updating the contents of the Switch Register. The Switch Bus is used to transfer the data from the Keyboard Display Register to the Switch Register. The data is multiplexed by digit over the bus to the Switch Register. This continually validates the Switch Register data. When the Switch Register data is to be cleared by striking the Clear Display key, the $\overline{\text{CD}}$ signal clears the Keyboard Display Register. The Switch Register becomes cleared as it is updated with the data from the Keyboard Display Register.

In controlling the numerical display, the Display Multiplexer circuit control the multiplexing of display data onto the Display Bus. The Multiplexing Clock oscillator (MUX CLK) and the Multiplex Phase signals (M1, M2, M3 and M4) are generated by the Display Multiplexer circuit for data synchronization. MUX CLK changes the phase signals cyclically from M1 through M4. All Display Bus data sources are controlled by these lines.

Enabling signals for the different Display Bus data sources are generated by this circuitry also. These four signal lines reflect the four display states of the front panel. These enabling signals are KEY, ENABLE D, ENABLE ADR and $\overline{\text{BLANK}}$. KEY indicates the displaying of Switch data from the Keyboard Display Register. ENABLE D causes the displaying of data from the D Register of the FPU. ENABLE ADR acts to enable the sourcing of Display Bus data from the Address Register of the FPU. $\overline{\text{BLANK}}$ disables the display. Of the four possible display states, the current state is indicated by the Display State Indicator. This is actually a two bit, modulo four ring counter. Its states are advanced by a Display Change ($\overline{\text{DX}}$) pulse. It can be set to different states by certain operational conditions.

When displaying Switch Register data, KEY is true. Data from the Keyboard Display Register is multiplexed onto the Display Bus. The keying of numerical data from the keyboard or the clearing of the display automatically sets the Display State Indicator to the KEY state.

ENABLE ADR causes data from the Address Register of the Control/Display System to be multiplexed onto the Display Bus. It can be automatically set to this state by the transition to a halt state by the processor. ENABLE D causes multiplexing of data onto the Display Bus from the D Register during phases M1 and M2. Since only two digits need be displayed in this case, $\overline{\text{BLANK}}$ is true during M3 and M4, when the D Register is displayed. The Display State Indicator is set to the $\overline{\text{BLANK}}$ state when the processor operation changes from halt to run. These are the only operational conditions where the Display State Indicator is set to a specific state. All other state changes occur when initiated by a DX keystroke.

FRONT PANEL CONTROL CIRCUIT

The Front Panel Control circuit, shown in Figure 8, controls the major flow of the front panel operation. It supplies controlling signals to the Front Panel Data circuit. In doing so, it synchronizes front panel miniprogram activity to processor activity.

Since the front panel functions are actually miniprograms stored in a memory in the Front Panel Data circuit, there must be a program counter used to correctly sequence through the routines. The Front Panel Control circuit contains an eight bit Micromemory Address Register used for this purpose. When initiating a front panel function, this register must be loaded with the correct starting address of the appropriate miniprogram. When the function is keyboard initiated, this address is the 8 bit code

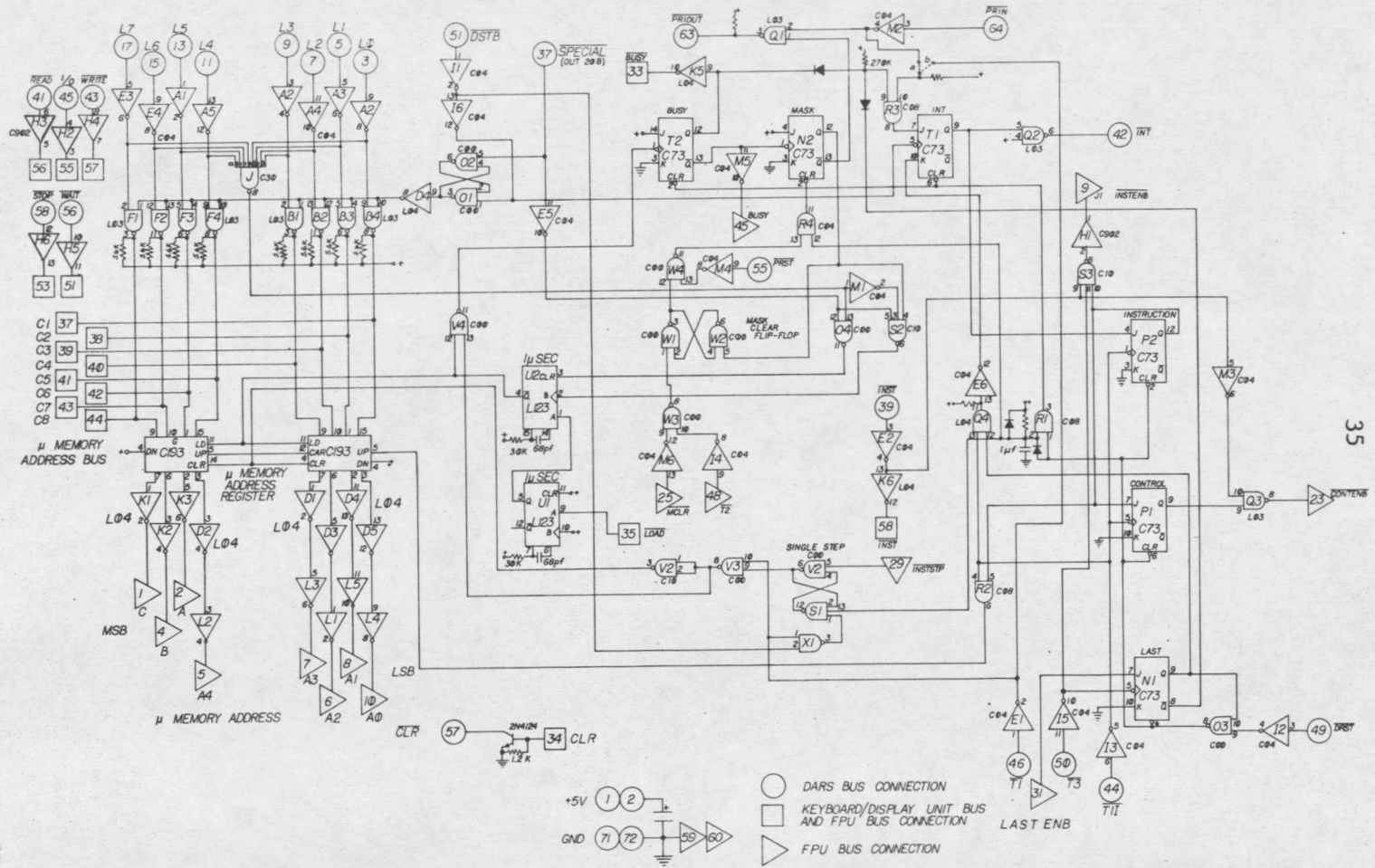


FIGURE 8: FRONT PANEL CONTROL SCHEMATIC

generated on the Display Multiplexer circuit and loaded from the Micro-memory Address Bus. When the front panel function is processor initiated, the data output from the processor's Accumulator by the special output instruction is the address stored in the Micromemory Address Register. This byte is sent via the microcomputer's L Bus. The $\overline{\text{LOAD}}$ signal causes the latching of the address from the Micromemory Address Bus when keyboard caused. The control line activated by an OUT 20B processor instruction contained in the DARS Bus causes the latching of the L Bus contents.

The latching of a new address into the Micromemory Address Register causes two function control flip-flops to be set, BUSY and MASK. BUSY indicates the FPU is executing an operation and inhibits a second operation from being started until the first is complete. MASK disables the interrupt priority out signal ($\overline{\text{PRIOUT}}$). This inhibits interrupt requests to the processor by interrupting devices of lower priority than the FPU. With the setting of these control signals, Control/Data System action is started.

A major function of the Front Panel Control circuit is to differentiate between processor instruction fetch cycles and other instruction cycles. During instruction fetch cycles, the FPU must furnish instructions to the processor. In all other cycles, the FPU manipulates data between itself and the DARS microcomputer system. To do this the Front Panel Control circuit uses two control signals, Instruction Enable

$\overline{\text{INSTENB}}$ and Control Enable ($\overline{\text{CONTENB}}$). $\overline{\text{INSTENB}}$, when true, indicates that the processor is in T3 of an instruction fetch memory cycle. $\overline{\text{CONTENB}}$, on the other hand, is valid during all other non-instruction fetch cycles. Both are valid only during front panel operations.

If the Front Panel has interrupt priority ($\overline{\text{PRIN}}$ being true), the setting of BUSY causes the front panel Interrupt Request flip-flop (INT) to be set. This causes an interrupt request on the bus. INT remains set until the final miniprogram instruction of the function. When the processor responds to the interrupt with timing control signal T11, the INSTRUCTION flip-flop of the Front Panel Control circuit is set. This indicates that miniprogram instructions may be accessed. It also enables the generating of $\overline{\text{INSTENB}}$, and the incrementing of the Micromemory Address Register at the beginning of each subsequent T11.

At the beginning of the second processor cycle after INSTRUCTION is set, the CONTROL flip-flop is set. This enables $\overline{\text{CONTENB}}$ during non-instruction fetch processor cycles.

During the execution of a front panel operation, the Front Panel Control circuit retains control of the processor and miniprogram execution. Processor control is maintained by the generation of processor interrupts. Control over the miniprogram is maintained by addressing through the Micromemory Address Register and the controlling of processor and front panel instruction interaction with $\overline{\text{INSTENB}}$ and $\overline{\text{CONTENB}}$. These processes continue for the duration of the front panel operation.

The termination of the front panel operation begins with the Last Enable control line (LASTENB) becoming true. LASTENB is generated by the Front Panel Data circuit. This sets the LAST flip-flop contained in the Control circuit. LAST indicates the following instruction is the final miniprogram instruction to be executed. LAST clears INT during T2 of the final instruction. It also resets BUSY, INSTRUCTION, CONTROL and LAST flip-flops during T3 of the last instruction. This terminates the front panel operation. Since these controlling flip-flops are clear in a processor cycle, the final instruction of the miniprogram should be a single cycle processor instruction.

The only flip-flop not cleared by LAST is MASK. The re-enabling of Priority Out ($\overline{\text{PRIOUT}}$) is done by a programmable clearing of MASK rather than automatically re-enabling it at the completion of the FPU operation. This is done so lower priority interrupts could be locked out from affecting the processor activity after completion of a front panel operation. This is useful when halting the processor and not enabling other interrupts which would cause it to restart.

MASK is cleared by the Priority Reset bus signal ($\overline{\text{PRST}}$) only after the FPU has set its MASK CLEAR flip-flop. This flip-flop is set by the Mask Clear ($\overline{\text{MCLR}}$) control line from the Front Panel Data circuit. When MASK is cleared, the MASK CLEAR flip-flop is also reset. The MASK also plays a determining role in the initiation of a FPU operation by the CPU in one instance. If the CPU initiates a front panel operation with

a starting address of zero (the conditional halt function), it is only executed if MASK is set.

The final function of the Front Panel Control circuit is its single instruction capability. The front panel is able to halt the processor after only one instruction has been executed. The FPU single step operation causes the Single Step flip-flop on the Control circuit to be set. It is set by the $\overline{\text{INSTSTP}}$ control line from the Data circuit. The T1 time period of the first non-FPU generated instruction causes the initiation of the halt operation of the front panel. When the CPU responds to the front panel's interrupt, after one instruction has been executed, the front panel halts the CPU's operation and resets the Single Step flip-flop.

FRONT PANEL DATA CIRCUIT

The Front Panel Data circuit, shown in Figure 9, operates under miniprogram control, initiated and controlled by the Front Panel Control circuit. It contains five data storage registers, a microcontrol memory and a microcontrol system. The five data registers are used for data storage by the FPU. The microcontrol memory contains the miniprograms directing the FPU operations. The microcontrol system directs data interaction between the front panel data registers and the DARS micro-computer system as per miniprogram instructions.

The five data storage registers are the Address Register, D Register, Switch Register, Save Register and Flag Register. The Address,

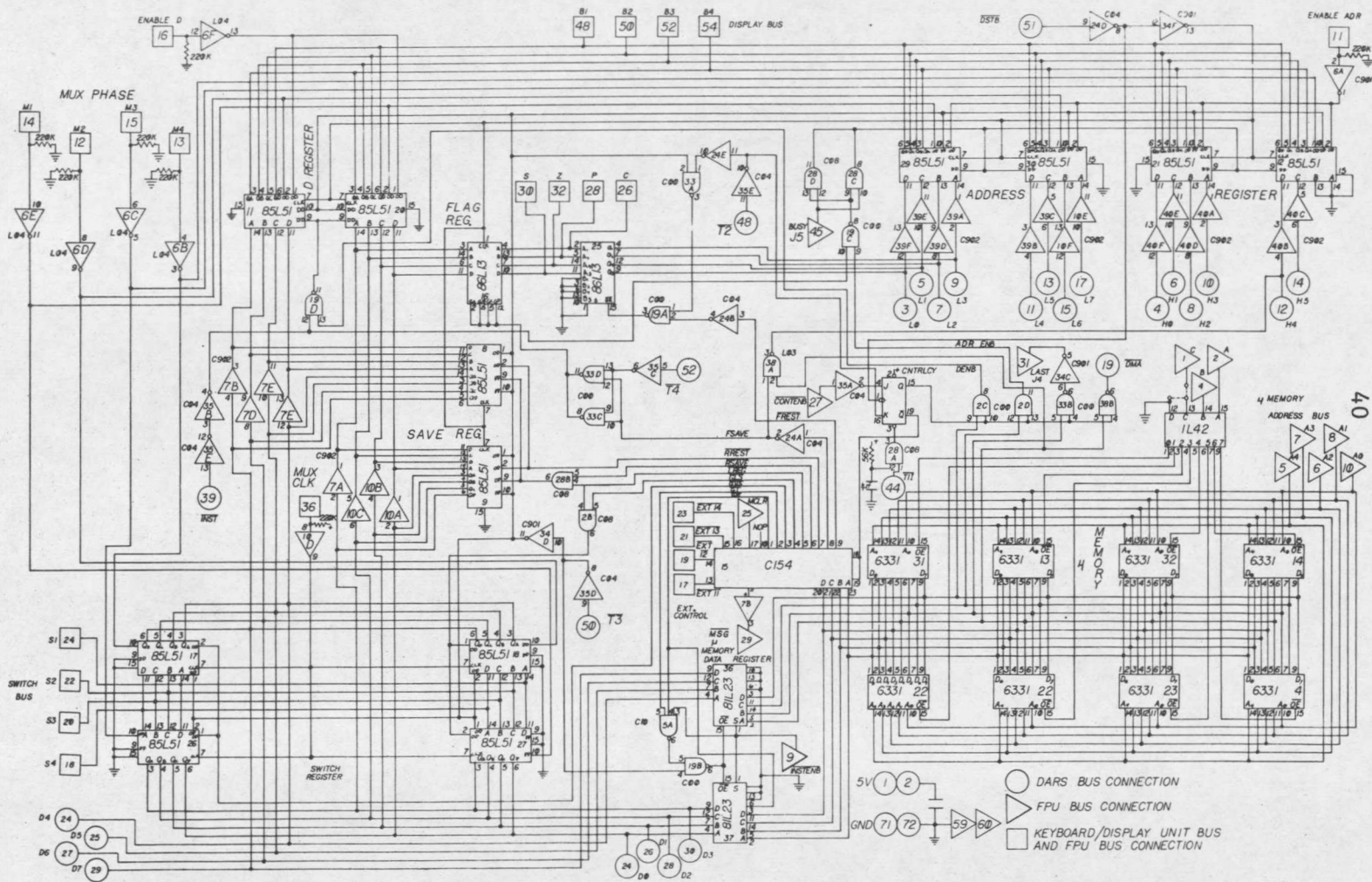
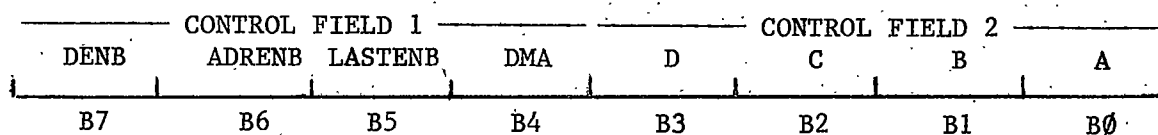


FIGURE 9: FRONT PANEL DATA SCHEMATIC

D and Switch Registers have been described in previous sections. The Save Register is used for single byte storage of data during FPU operations. The Flag Register is used for storing the four processor flag bits. This is used for the Flag status indicators and for retention of processor flag status. It is possible for flag states to change during front panel operations, so initial flag states are saved and can be re-stored when the front panel operation is completed.

The microcontrol memory contains 256 bytes of Read Only Memory. All front panel miniprograms are stored here. One byte of data is accessed from the microcontrol memory each processor cycle during an FPU operation. This memory is organized into 8 pages of 32 bytes each. All page selection and byte addressing is done by the Micromemory Address Register. During instruction fetch cycles, the byte accessed is sent to the processor over the D Bus. It is a standard processor instruction. During all other cycles, the byte is used for data manipulation with the microcontrol system, and are called microcontrol instructions.

Front panel microcontrol instructions are different from the standard processor instructions. There are twenty different operations that can be done by the microcontrol system, and up to five operations may be encoded into one microinstruction for simultaneous operation. The format of the microinstruction is shown in Figure 10. There are two control fields in the microinstruction, Control Field 1 and Control Field 2. The four bits in Control Field 1 (A, B, C and D), define one



MICROINSTRUCTION FORMAT

FIGURE 10

of 16 control operations. These operations are shown in Table 2 along with their bit configurations and functions. Control Field 2 defines the four remaining microcontrol operations. A bit set in the appropriate Control Field 2 position activates the respective operation. All data manipulation capability of the Front Panel Data circuit is exercised through these microinstructions.

TABLE 2

CONTROL FIELD 1 FUNCTIONS

				HEX		
D	C	B	A	FORMAT	NAME	FUNCTION
0	0	0	0	0	G0	Gate binary zero onto bus
0	0	0	1	1	G1	Gate binary one onto bus
0	0	1	0	2	GHS	Gate high switch data onto bus
0	0	1	1	3	GLS	Gate low switch data onto bus
0	1	0	0	4	LREG	Load Save Register from switches
0	1	0	1	5	RSAB	Load Save Register from bus
0	1	1	0	6	RRST	Restore bus data from Save Reg.
0	1	1	1	7	FSAB	Save Flag status in Flag Register
1	0	0	0	8	FRST	Restore Flag status to bus
1	0	0	1	9	NOP	No operation
1	0	1	0	A'	INSTP	Instruction step
1	0	1	1	B'	EXT11	External Line 11 (Message Lamp)
1	1	0	0	C'	EXT12	External Line 12
1	1	0	1	D'	EXT13	External Line 13
1	1	1	0	E'	EXT14	External Line 14
1	1	1	1	F'	MCLR	Clear MASK flip-flop enable

Control Field 1 Instructions

The 16 operations encoded in Control Field 1 are as follows. Gate \emptyset ($G\emptyset$) causes a binary zero to be placed on the D Bus during T3 of the present cycle. Gate 1 ($G1$) similarly causes the gating of a binary one. Gate High Switch Data (GHS) gates the high order eight bits of the Switch Register data onto the D Bus during the time period T3. Gate Low Switch Data (GLS) similarly gates the low order eight bits of the Switch Register bits both onto the D Bus and into the Save Register at T3. Register Save ($RSAV$) causes the contents of the D Bus to be stored into the Save Register at the current T3. Register Restore ($RRST$) reverses the operation of $RSAV$ and places the contents of the Save Register onto the D Bus during T3. Flag Save ($FSAV$) causes the four low order bits of the D Bus to be gated into the Flag Register during T4. This must be done during an I/O cycle of an Input instruction as this is the only time flag values appear on the D Bus. Flag Restore ($FRST$) causes the four bits of the Flag Register to be gated onto the lower four bits of the L Bus. To restore the flag condition, these flag bits are used to index a table in the top 16 locations of memory. These are ROM locations which contain constants that when added to themselves causes the flags to take on the same state of the flags when they were stored. When the flag bits are gated onto the L Bus, the other address bits on the H-L Bus are in a high state, if DMA of Control Field 2 is active, and the appropriate locations are addressed. No Operation (NOP) causes no

control action to be taken with Control Field 1. The four external control signals are used for control signals outside of the Front Panel Data circuit. EXT11 is used to set the Message Indicator flip-flop on the Front Panel Display Multiplexer circuit. The three remaining external control lines are unused. The final control line in Control Field 1 is the Mask Clear (MCLR). As explained earlier, it allows the MASK flip-flop to be reset by a PRST instruction from the processor.

Control Field 2 Instructions

Although only one Control Field 1 function may be active in any one microinstruction execution, any, all or none of the functions of Control Field 2 may be executed in one microinstruction execution. Disable Memory Address (DMA) disconnects the H-L Bus from CPU control. This causes a condition on the bus where all bits are at a high level. The Last Enable (LASTENB) line, as explained earlier, causes the LAST flip-flop on the Front Panel Control circuit to be set, terminating the front panel operation. The Address Enable (ADRENB) function causes the Address Register to store the contents of the H-L Bus during T3. The D Enable control bit (DENB) causes a similar operation to occur with the D Bus and the D Register during T3.

Appendix II lists the instructions of the miniprograms controlling the front panel operations. The processor's Accumulator is considered a scratch register during most front panel operations. In all operations except RUN, HALT, LDA and DA the contents of the Accumulator are lost.

However, HALT stores the contents of the Accumulator in the Save Register as the operation starts. RUN loads the Accumulator with the contents of the Save Register. DA displays the contents of the Save Register. LDA stores the lower byte of the Switch Register in the Save Register. With proper use of these operations, the Accumulator's contents can be saved during front panel operations.

Displaying Data Register Contents

The Address Register and D Register connect to the Display Bus. When these registers are enabled by either ENABLE D or ENABLE ADR, their contents are multiplexed onto the Display Bus. This multiplexing is done and is controlled by the MUX PHASE control lines. In this manner, the display can show the contents of either of these registers. During CPU program operation, the Address Register is constantly being updated by the contents of the program counter. Front panel routines which change this action are LDHL, DHL, LNM and DNM. These cause the Address Register to contain the contents of the H and L processor registers. On the other hand, the D Register only contains data specifically stored there by FPU operations and does not change with normal CPU activity.

Although the Switch Register is not directly displayed, it is continually updated by data from the Keyboard Display Register through the Switch Bus. This enables it to be identical to the displayed value.

SUMMARY

The Front Panel Unit is a keyboard and display centered device designed to be the primary interface between the DARS system and its operator. It is the tool by which the operator examines and/or modifies the system operation. It also functions as an input medium for operator generated data. Although primarily used by operator initiated actions, it also can be initiated by the processor of the microcomputer system.

It is structured to use the micorprocessor to accomplish the data manipulation tasks. It utilizes miniprograms to control processor activity and interacts with the processor as it executes these programs to manipulate or sample data. It is designed to operate in this manner to minimize burden on the processor in displaying data and initiating various necessary operations.

The structure of the FPU consists of two major systems, a Keyboard/Display System and a Control/Data System. The Keyboard/Display System is used by the operator to initiate front panel activity and to examine front panel information. The Control/Data System does the actual front panel-processor interaction. This system controls and interacts with the processor during the operations. It contains the data and control capability for all front panel operations.

CHAPTER 6

DATA ACQUISITION SYSTEM

In data acquisition, input data is monitored from points external to the DARS unit. These monitoring points sense certain parameters of the system for analysis. These parameters may be temperature, force, speed, specific events, etc. It is the function of the transducer to change these quantities into electrical signals. These electrical signals, in turn, must be converted to a digital format for the DARS CPU. Data acquisition interfaces perform these functions.

It is desirable in data acquisition systems that numerous parameters be monitored simultaneously. Also, each monitoring point should be individually accessible. In DARS therefore, provisions were made to have a separate data bus from one I/O port to which all monitoring units are tied for control, status and data transfer purposes. This is the Data Acquisition System (DAS) portion of DARS.

An important aspect when monitoring data in an atmospheric environment is the protection of the system from electrostatic discharges. It is necessary to protect the system so that if the system were to fail as a result of high voltage, power loss, short circuit, etc., the remaining network would function.

In the design of the DARS system, was included a data bus electrically isolated from the I/O port and the processing system. The control, status and data lines of the I/O port were duplicated in the DAS data bus. All the transducer system (transducers and interface electronics)

is separate from the controlling portion of DARS.

The Data Acquisition System is designed to contain different interface modules. These modules are connected to the DAS data bus. Each module may be activated separately to furnish desired data to the I/O port when polled by the processor. Each module responds to a separate address. When the module detects its particular address as a portion of the data output from the port, it is activated. In this manner, each transducer interface is a separate entity. However, control from the DARS CPU acts on the entire DAS bus as a whole. The External Control lines from the I/O port connect in a parallel manner to each module, so the control lines affect the system in its entirety. Currently, only one control line is used. EC4, $\overline{\text{CLR}}$, acts as the bus enabling signal. To perform any function, $\overline{\text{CLR}}$ must be false, as it holds, when in a true state, all modules in a nonfunctional or cleared mode.

ISOLATION SYSTEM

Electrical isolation between the I/O port and the DAS data bus is achieved by using devices known as optical isolators. An optical isolator is a unit which uses light rather than electrical signals to transmit data. It is comprised of a light emitting diode (LED) and a photo transistor. Light is generated by the LED when current flows through it, and is detected by the photo transistor, turning it on. Otherwise, the transistor is off when the LED is not emitting light. Using the binary bus data to control the LED's, data can be transmitted between

two systems.

Referring to the Isolation System schematic, Figure 11, data and control signal lines from the I/O port and the DAS bus drive circuitry to turn off or on the LED's in the optical isolators. The off or on state of the photo transistors are then buffered to drive the appropriate bus or port lines.

The Isolation System primarily acts as a signal buffer between the I/O port and the DAS data bus. However, signal delays can occur during signal transistions due to the relatively slow nature of the optical isolators. To compensate for this, the two data ready signals, Input Ready (INPRDY) and Output Ready (OUTRDY), are manipulated by the Isolation System. It delays these signals in order for the respective levels on the data lines to stabilize. In this manner, no erroneous data can result from the transition of a ready line before all transitions on the data lines are complete.

These delays are generated by one-shot pulse generators. When a change of state on a ready line occurs, it triggers a delay pulse. The trailing edge of the delay pulse causes the isolated ready line to change state. When the ready line that had initiated the state change returns to its normal state, the isolated ready line immediately follows with no delay. Figure 12 shows this sequence in a timing diagram. Note, too, in addition to the delay, there is a sense inversion on the ready line. $\overline{\text{INPRDY}}$ and $\overline{\text{OUTRDY}}$ of the I/O port are $\overline{\text{INPRDY}}$ and $\overline{\text{OUTRDY}}$ of the DAS bus.

