



This is to certify that the

thesis entitled

A Minicomputer-Based Automatic
Real-Time Data Acquisition System

presented by

Walter Nester II

has been accepted towards fulfillment
of the requirements for

Master of Science degree in Electrical

Engineering and Systems Science

A handwritten signature in cursive script, appearing to read "P. A. Fisher", written over a horizontal line.

Major professor

Date 2/22/78

**A MINICOMPUTER-BASED AUTOMATIC
REAL-TIME DATA ACQUISITION SYSTEM**

By

Walter Nester II

A THESIS

**Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of**

MASTER OF SCIENCE

**Department of Electrical Engineering
and Systems Science**

1978

2112541

ABSTRACT

A MINICOMPUTER-BASED AUTOMATIC REAL-TIME DATA ACQUISITION SYSTEM

by

Walter Nester II

A minicomputer-based data acquisition system which automatically records data from remotely located groups of sensors has been designed, constructed, and tested. It was developed to meet the data gathering requirements of the Water Quality Management Facility of the Institute of Water Research at Michigan State University but has general applicability to other large-scale, distributed, physical systems. The minicomputer communicates with each group of sensors via dial-up telephone lines. Each sensor is automatically sampled at a fixed rate; the maximum sampling rate is once every fifteen minutes. Once a day the minicomputer transfers the accumulated sensor data to a large, main computer, via a dial-up telephone line, where it is entered into a permanent data base. A Real-Time Operating System also allows an operator to simultaneously access the data acquisition system to obtain the current value of any sensor.

ACKNOWLEDGEMENTS

I would like to thank the following people for their support on this project: Gary S. Krause for designing and constructing portions of the Remote Station which was used to test the DAS in the laboratory; Sigurd L. Lillevik for his floppy disk-based object file loader program, SIGMA, which greatly simplified the software development phase of this project; and a special thanks to my advisor and major professor, Dr. P. D. Fisher. I am grateful for his continued encouragement, assistance, and patience throughout this project.

TABLE OF CONTENTS

Chapter 1	INTRODUCTION	1
Chapter 11	HARDWARE ORGANIZATION	6
2.1	Introduction	6
2.2	Central Station Organization	8
2.2.1	Minicomputer	8
2.2.1.1	Processor	8
2.2.1.1.1	Power Fail Detector	11
2.2.1.1.2	Real-Time Clock	11
2.2.1.1.3	Teletype Port	11
2.2.1.2	16K Memory.	12
2.2.1.3	Disk Controller	12
2.2.1.4	Automatic Calling Unit Multiplexer (ACUM)	12
2.2.1.5	Asynchronous Modem Controller (ACM)	12
2.2.1.6	Custom Controller	13
2.2.2	Dual Floppy Disk Unit	13
2.2.3	Modem/Calling Units	15
2.2.4	Display Controller.	16
2.2.5	Local Modem/Controller.	19
2.2.6	Battery Real-Time Clock	22
2.2.7	Remote Modem/Controller	24
2.3	Remote Station Organization	25
Chapter 111	SYSTEM ORGANIZATION.	31
3.1	Introduction.	31
3.2	Automatic Acquisition Mode.	31
3.3	Dump Mode	33
3.4	Disk Error Mode	35
3.5	Power Restart Mode.	35
3.6	Manual Mode	36
3.6.1	A Command	37
3.6.2	D Command	38
3.6.3	H Command	38
3.6.4	K Command	39
3.6.5	P Command	40
3.6.6	S Command	41
3.6.7	T Command	42
3.6.8	U Command	42
3.6.9	V Command	43
3.6.10	W Command.	44
3.6.11	X Command.	45
3.6.12	Y Command.	48

Chapter IV	SOFTWARE ORGANIZATION	49
4.1	Introduction.	49
4.2	Interrupts.	49
4.3	Station Definition Blocks	50
4.4	Sensor Definition Blocks.	51
4.5	Floppy Disk Organization.	51
4.6	Real-Time Clock	53
4.7	Automatic Acquisition Mode.	56
4.7.1	Error Message Print Routine	64
4.8	Dump Mode	70
4.9	Disk Error Mode	79
4.10	Power Restart Mode	84
4.11	Manual Mode.	89
4.11.1	A Command.	96
4.11.2	D Command.	106
4.11.3	H Command.	107
4.11.4	K Command.	110
4.11.5	P Command.	111
4.11.6	S Command.	116
4.11.7	T Command.	117
4.11.8	U Command.	118
4.11.9	V Command.	118
4.11.10	W Command	120
4.11.11	X Command	121
4.11.12	Y Command	124
Chapter V	CONCLUSION	126
5.1	Objectives.	126
5.2	Accomplishments	126
5.3	Overall Project Evaluation.	130
Appendix A	BLOCK DIAGRAM COMPOSITION	132
Appendix B	DISK ERROR CODES.	136
Appendix C	BASE PAGE INTERRUPT LOCATIONS	137
References.		139

LIST OF TABLES

5.1	Cost Breakdown	.129
A.1	Custom Controller Block Diagram IC Composition	.132
A.2	Local Modem/Controller Block Diagram IC Composition. .	.133
A.3	Display Controller Block Diagram IC Composition. . . .	.134
A.4	Battery Real-Time Clock Block Diagram IC Composition .	.134
A.5	Remote Modem/Controller Block Diagram IC Composition .	.135

LIST OF FIGURES

1.1	Water Quality Management Facility - Lake System.	2
2.1	Data Acquisition System Block Diagram.	9
2.2	Central Station Block Diagram.	10
2.3	Custom Controller Block Diagram.	14
2.4	Display Controller Block Diagram	17
2.5	Local Modem/Controller Block Diagram	20
2.6	Battery Real-Time Clock Block Diagram.	23
2.7	Remote Modem/Controller Block Diagram.	26
2.8	Remote Station Block Diagram	27
3.1	Floating Point Format.	48
4.1	Real-Time Clock Flow Chart	57
4.2	Automatic Acquisition Mode Flow Chart.	65
4.3	Dump Mode Flow Chart	80
4.4	Disk Error Mode Flow Chart	85
4.5	Power Restart Mode Flow Chart.	90
4.6	Manual Mode Flow Chart	97

CHAPTER I

INTRODUCTION

As a prerequisite to understanding and subsequently controlling a physical process, a suitable data base is required. And many physical systems are of the large-scale, distributed type, in which the individual points at which data must be gathered are widely separated, e.g. manufacturing or ecological systems. Thus a problem exists in gathering the data and making it available for the decision-making process. This thesis describes the hardware and software design and the operational characteristics of a real-time data acquisition system which solves these data gathering problems as they apply to a specific large-scale distributed system; namely, the Water Quality Management Facility of the Institute of Water Research at Michigan State University.

This Facility is used for the study of the restoration of water quality by natural processes to secondary effluent from the East Lansing sewage treatment plant.⁽¹⁾ This research is being conducted at a four-lake system on the Michigan State University campus (see Figure 1.1). Effluent enters lake 1 and progresses sequentially to lake 4 by gravity feed. Nutrients are removed from the waste water as it proceeds through the system by incorporating them into harvestable aquatic plants. The water emerging from lake 4 is then used to spray-irrigate crops. It is desired to periodically measure and record chemical properties of the water, such as temperature,

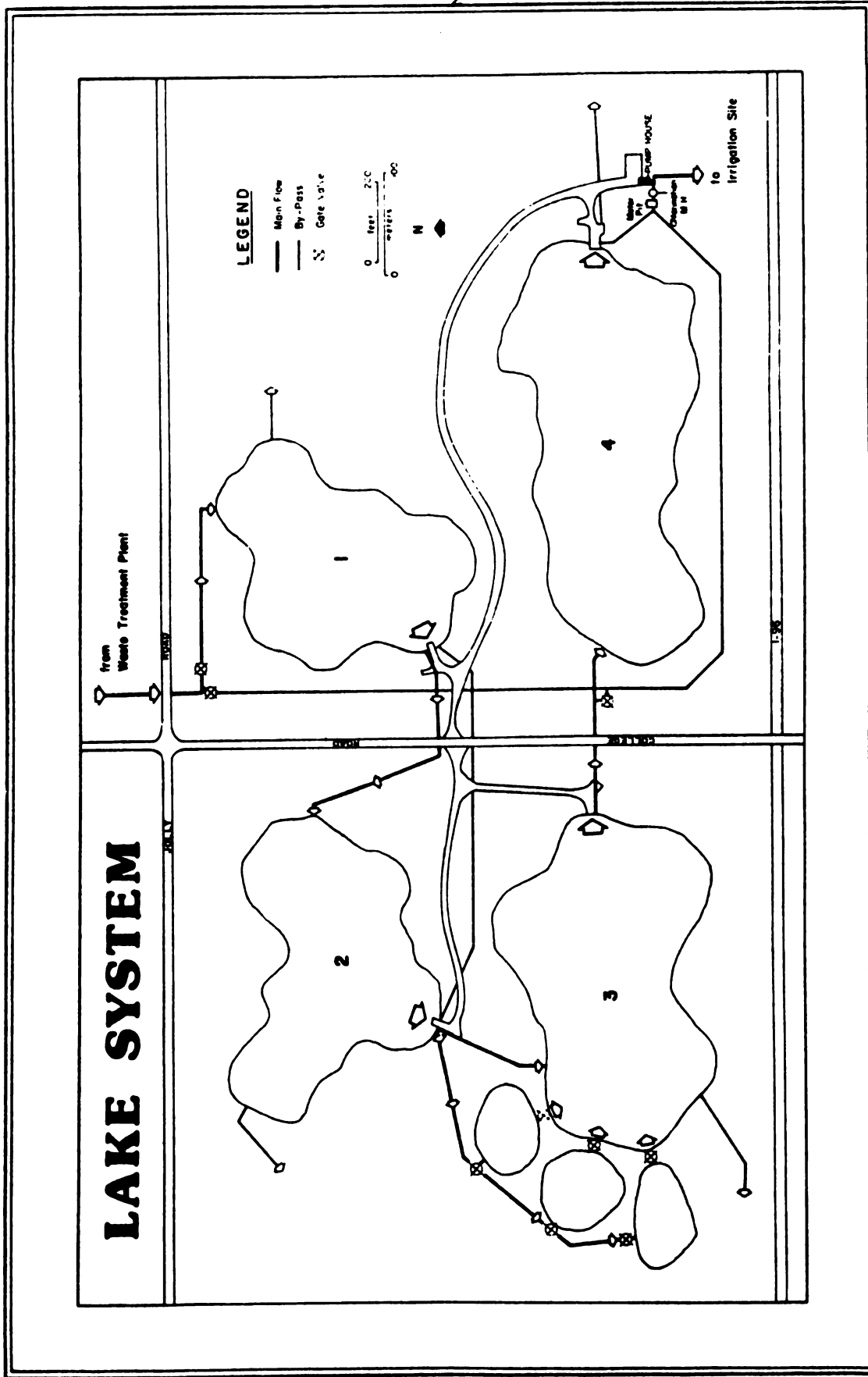


Figure 1.1 Water Quality Management Facility - Lake System

dissolved oxygen content, turbidity, pH, and conductivity, at the outlets of each of the four lakes. In addition environmental parameters such as wind speed, wind direction, solar intensity, air temperature, dew point, and soil flux must also be monitored. These sensors are widely distributed throughout the Water Quality Management Facility.

Several methods of collecting data from such widely-distributed sensors are in common use. A mechanical recording device, e.g. stripchart recorder or mechanical printer, can be located near each sensor. There are several disadvantages to this approach, however. The printed output from each of the recording devices must be manually picked up and taken to a central location for collation and preliminary analysis. Thus, the value of the physical parameter cannot be determined until the printed record has been collected and examined. This time lag in obtaining data makes it difficult to make operational decisions based on current data. The recording devices themselves could be centrally located and connected to the remote sensors via individual communication channels. However, the cost of these communication channels would be prohibitive. Further, the printed data must still be manually entered into a computer for numerical analysis. This last problem may be remedied by including a paper tape punch and suitable interface equipment at each sensor. The paper tapes could then be read into the computer, eliminating manual data entry. However, the paper tapes must still be picked up from each sensor location, and the cost of installing a paper tape punch at each sensor would be prohibitive.

There is often a more efficient method to collect such data. Several sensors, located near each other, may be connected to a

4

single communication channel. The communication channels from several such groups of sensors may then be connected to a centrally located data logger. This data logger may interrogate each sensor at fixed intervals of time and temporarily store the sensor's data. Once a day the data logger may then call up a large main computer on a dial-up telephone line and send it the accumulated sensor data. Researchers can then easily retrieve the data from this main computer for analysis. Using this method, no human intervention is necessary to sample the sensors and transfer the data to the main computer. This results in a great savings in human time and effort. Additionally, such a data acquisition system allows an operator to quickly obtain the current value of any sensor; thus, operational decisions can be based on current data.

The design objectives for such a Data Acquisition System (DAS) for the Water Quality Management Facility are: data from a maximum of 16 widely-separated groups of sensors must be automatically recorded at fixed intervals of time; the sampling rate for each sensor must be individually specified, with a maximum sampling rate of once every 15 minutes; the accumulated data must be automatically transferred to the main campus computer once a day, over a dial-up telephone line, and held at the DAS if the computer is unavailable for up to six days; the DAS must correctly restart itself after a power failure has occurred; data returned from each sensor must be checked against specified tolerance limits, and an error message printed if the data exceeds these limits; and, a non-technically trained operator must be able to access the DAS, from either the location of the data logger or from the Water Quality Management Facility in order to obtain a printed listing of the data

automatically recorded from a specified sensor during the previous 24 hours, obtain the current value of any sensor, define a new sensor, remove an existing sensor, and check the status of the DAS.

Chapter 2 of this thesis describes the hardware organization of such a data acquisition system. Chapter 3 explains the overall operation, including the interactive commands which allow an operator to access the DAS. Chapter 4 details the software organization and the Real-Time Operating System. Chapter 5 concludes with a summary of the performance of the DAS to date and suggestions for possible improvements.

CHAPTER II

HARDWARE ORGANIZATION

2.1 Introduction

The DAS is configured as a Central Station and at least one Remote Station. The Central Station supervises the data acquisition process, while the Remote Stations interface to the sensors. The Central Station can support a maximum of 16 Remote Stations.

Each Remote Station can be connected to a maximum of 62 sensors; 31 of the sensor inputs accept an analog voltage between 0 and 10 VDC. The remaining 31 sensor inputs accept a 12-bit digital TTL input. Two additional inputs are included for diagnostic purposes. Each Remote Station is also connected to a dial-up telephone line. When the Central Station wants to sample a sensor at a Remote Station, it first dials the telephone number of that Remote Station. Thus, a Remote Station can be located anywhere a direct-dial telephone line is available. The Remote Station detects the ringing signal and takes its telephone off-hook. The Central Station then sends five ASCII characters to the Remote Station to indicate which sensor should be sampled. The Remote Station decodes the characters and selects the correct sensor input channel. If one of the analog channels was selected, that input is converted to a 12-bit digital number. No conversion is necessary if one of the digital channels was selected. The Remote Station then codes the 12 bits of sensor data into five ASCII characters and sends these characters back to

the Central Station. The Central Station may then request the Remote Station to sample another sensor or may terminate the telephone connection. When the Remote Station detects that the telephone connection is broken, it puts its telephone on-hook and waits for the next call.

The Central Station automatically dials up each Remote Station when a sensor at the Remote Station must be sampled. Each sensor is automatically sampled by the Central Station at the interval of time specified when the sensor was initially defined. The data from each sensor is stored on a floppy disk at the Central Station. Every weekday this accumulated sensor data is transferred to the main campus computer, a Control Data Corporation 6500. The Central Station, via one of its two telephone lines, dials up the 6500 and sends it the sensor data in a condensed format. The Central Station then commands the 6500 to execute a program which converts the data back to its original form and adds it to the permanent data base. The sensor data can now be easily accessed on the 6500 Computer system. The Central Station also allows an operator access to the DAS via two Teletypes. The Local Teletype is located adjacent to the Central Station rack cabinet, while the Remote Teletype is located in the Pump House at the Water Quality Management Facility. The Remote Teletype allows an operator to gain access to the DAS from a location where data is being gathered. An operator, using either Teletype, may obtain the current value of any sensor. The Central Station automatically dials up the correct Remote Station and requests it to sample the designated sensor. The correctly scaled and labeled value of the physical parameter is then printed on the Teletype. The Central Station can also list the data from a given sensor when it

8

was automatically sampled during the previous 24 hours. An operator may also add or remove sensors and stations via one of the Teletypes. Thus the DAS can be easily modified to accomodate a changing sensor network.

A block diagram of the complete DAS is shown in Figure 2.1; Appendix A contains tables which list the integrated circuits that are contained in each functional block on the block diagrams for the Custom Controller, the Display Controller, the Local Modem/Controller, the Battery Real-Time Clock, and the Remote Modem/Controller.

2.2 Central Station Organization

The Central Station consists of a rack cabinet, two data couplers, and two Teletypes. The rack cabinet contains a minicomputer, display panel, dual floppy disk unit, auxiliary chassis, and two modem/calling units. A block diagram of the Central Station is shown in Figure 2.2; the function of each component is explained in the following sections.

2.2.1 Minicomputer

The minicomputer is a Computer Automation LSI-2/20, a general-purpose machine which operates with a 16-bit word length. Each block on the left-hand side of Figure 2.2 is a separate circuit card in the minicomputer chassis. The function of each of these cards is explained in the following subsections.

2.2.1.1 Processor

The processor is responsible for executing each instruction of the Real-Time Operating System. It also controls all interaction with the peripheral equipment. The processor card contains the Power

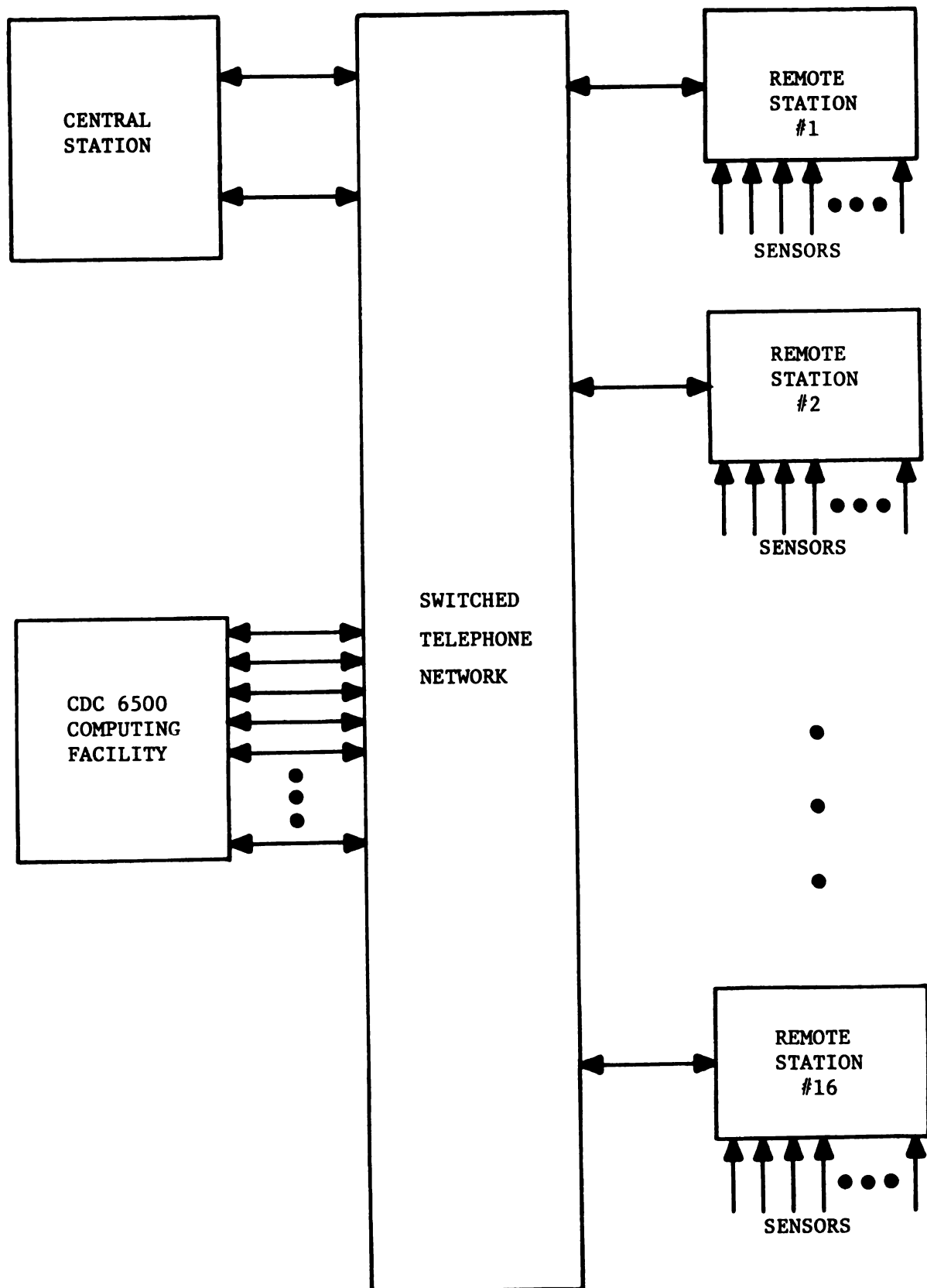


Figure 2.1 Data Acquisition System Block Diagram

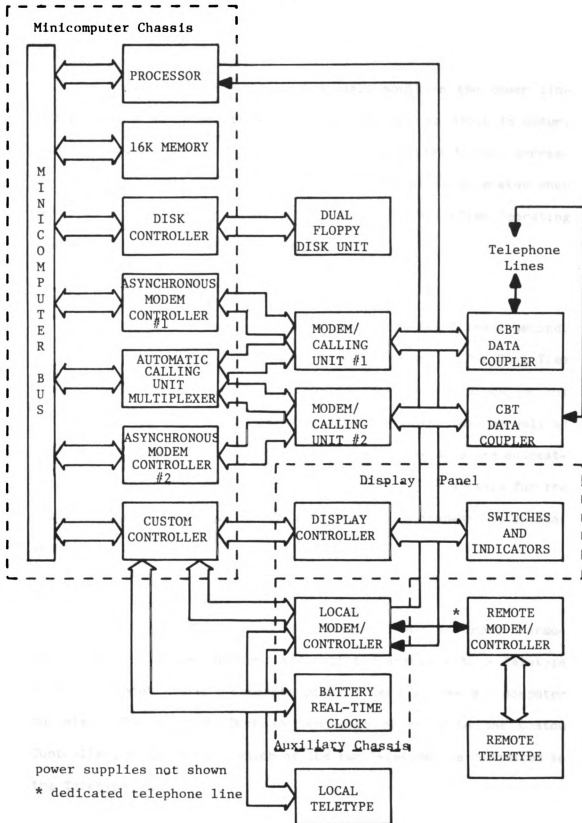


Figure 2.2 Central Station Block Diagram

11

Fail Detector, the Real-Time Clock, and the Teletype port.

2.2.1.1.1 Power Fail Detector

The Power Fail Detector continuously monitors the power line and generates an interrupt when a power failure is about to occur. This interrupt causes the Real-Time Operating System to halt processing and shut down the DAS. A second interrupt is generated when power is restored to an acceptable level. The Real-Time Operating System then begins restarting the DAS.

2.2.1.1.2 Real-Time Clock

The Real-Time Clock generates an interrupt 120 times a second, in synchronization with the power line frequency. The Real-Time Operating System uses this interrupt to maintain a clock in the software. This clock keeps track of the time of the day, as well as the day of the year, and determines exactly when sensors are automatically sampled. The power line frequency is used as a basis for the clock because of its superior long-term accuracy compared to crystal oscillators.

2.2.1.1.3 Teletype Port

The Teletype (TTY) Port allows the processor to input characters from a Teletype Keyboard and output characters to a Teletype printer without using a separate circuit card in the minicomputer chassis. The Teletype Port is connected directly to the Custom Controller, which selects which of the two Teletypes is connected to the Teletype Port.

2.2.1.2 16K Memory

The 16K Memory is organized as 16,384 words. The Real-Time Operating System itself is stored in the magnetic cores which make up the memory. Core memory is used, rather than semiconductor memory, so that the memory's contents will not be destroyed during a power failure.

2.2.1.3 Disk Controller

The Disk Controller interfaces the dual floppy disk unit to the minicomputer. The Disk Controller uses Direct Memory Access (DMA) to store data from core memory onto one of the disks or to transfer data from a disk to core memory.

2.2.1.4 Automatic Calling Unit Multiplexer (ACUM)

A single ACUM card is connected to both modem/calling units. The ACUM, under control of the processor, instructs a calling unit to initiate a telephone call, and then supervises the dialing of the telephone number. The ACUM also generates interrupts to indicate to the processor the progress of the call.

2.2.1.5 Asynchronous Modem Controller (AMC)

Two AMC cards are installed in the minicomputer chassis. Each AMC is connected to one of the modem/calling units. After a calling unit has dialed a telephone number and established a data connection, the AMC supervises the flow of data to and from the associated modem. Each AMC is set up to transmit and receive at 300 baud, with even parity. The Remote Stations and the 6500 Computer system operate with this same data convention.

2.2.1.6 Custom Controller

The Custom Controller functions as an interface between the minicomputer and the battery-operated Real-Time Clock, the Display Panel, and the two Teletypes. It decodes certain input, output, and select instructions and sends control signals to the Local Modem/Controller, the Battery Real-Time Clock, and the Display Controller. The Custom Controller also selectively transmits the signals on the minicomputer data bus to the Battery Real-Time Clock and the Display Controller, and receives signals from the Battery Real-Time Clock which it then selectively transmits onto the minicomputer data bus.

The Custom Controller also responds to three sense instructions; it selects the proper sense line and connects it to the minicomputer bus when the sense instruction is being executed:

SEN :13,1 - checks the status of the PRINT ERRORS switch on the Local Teletype. If the switch is on, the next sequential instruction is skipped.

SEN :13,2 - checks the status of the Manual Mode Authorization Override switch on the rear panel of the Central Station rack cabinet. If the override switch is on, the next sequential instruction is skipped.

SEN :15,0 - checks the status of the RESET push button switch on the Display Panel. If the button is pushed in, the next sequential instruction is skipped.

A block diagram of the Custom Controller is shown in Figure 2.3.

2.2.2 Dual Floppy Disk Unit

The Dual Floppy Disk Unit contains two floppy disk drives and an interface which connects them to the Disk Controller. This interface

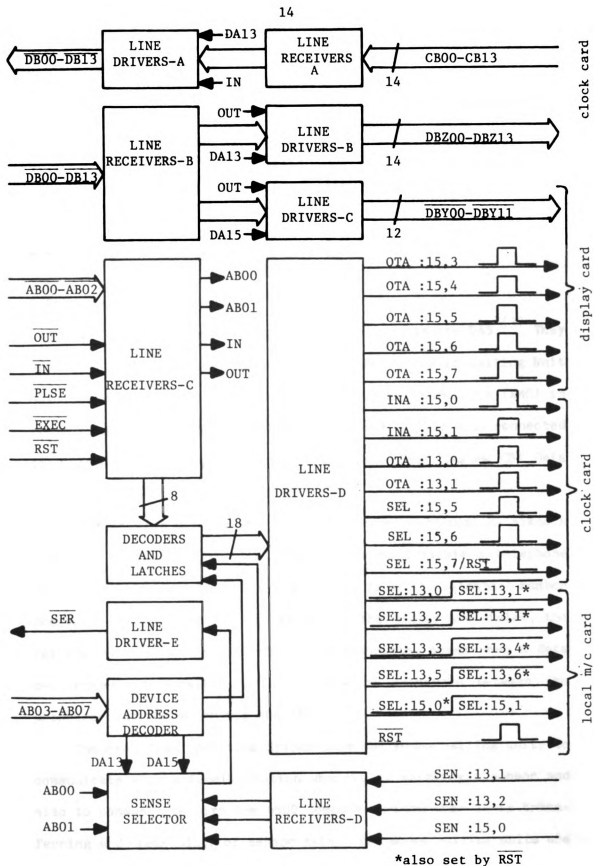


Figure 2.3 Custom Controller Block Diagram

has been modified so that neither disk can be write-protected. This modification is necessary since the disks normally become write-protected when power is initially applied,⁽²⁾ and the Real-Time Operating System must always be able to write on both disks.

The DAS stores the automatically-gathered data from each sensor on the disks. The disks also contain the parameters which define each sensor. Chapter 4 gives the locations on each disk where the sensor data and definitions are stored.

2.2.3 Modem/Calling Units

Two Vadic modem/calling units are included in the DAS.⁽³⁾ They are interfaced to the minicomputer thru the Automatic Calling Unit Multiplexer (ACUM) and two Asynchronous Modem Controllers (AMC) in the minicomputer chassis. Each modem/calling unit is also connected to a separate dial-up telephone line via a Bell System CBT Data Coupler.⁽⁴⁾

Each calling unit, when it receives the correct control signals from the ACUM, instructs its Data Coupler to initiate a telephone call. The calling unit then dials the digits of the telephone number presented to it by the ACUM. After all the digits are dialed, the calling unit transfers control to the associated modem when a data set answers the call. Data can now be exchanged between the processor and the modem via the AMC.

The Real-Time Operating System uses the modem/calling units to communicate with a Remote Station when it is sampling a sensor and also to communicate with the 6500 Computer system when it is transferring a 24 hour block of sensor data. Two modem/calling units are included so that Remote Stations can be called at the same time that

16

data is being transferred to the 6500.

2.2.4 Display Controller

The Display Controller interfaces the Display Panel numeric readouts, indicators, and switches to the Custom Controller. The Custom Controller signals the Display Controller when certain output instructions are executed by the processor and transfers the data on the minicomputer data bus to the Display Controller.

Numeric readouts on the Display Panel indicate the DAS time (hours, minutes, and seconds) and the day of the year. The status of the DAS is shown via three indicators: RUN, START, and FATAL ERROR. The green RUN indicator shows that the DAS is functioning normally; the yellow START indicator shows that the DAS is recovering from a power failure; and the red FATAL ERROR indicator shows that a disk error has occurred. The Display Controller also turns on a loud audible alarm and the red ALARM STOP indicator when the Custom Controller signals that a Fatal Error has occurred. When the flashing red ALARM STOP indicator is pushed in, the Display Controller turns off the alarm. The Display Controller signals the Custom Controller when the blue RESET push button on the Display Panel is pushed in. A block diagram of the Display Controller is shown in Figure 2.4.

The following instructions affect the operation of the Display Controller after they are decoded by the Custom Controller:

OTA :15,3- stores bits 0 thru 3 of the A register in a latch on the Display Controller card. The bits have the following significance: bit 0 controls the RUN indicator lights on the Local Teletype and the Display Panel, and when

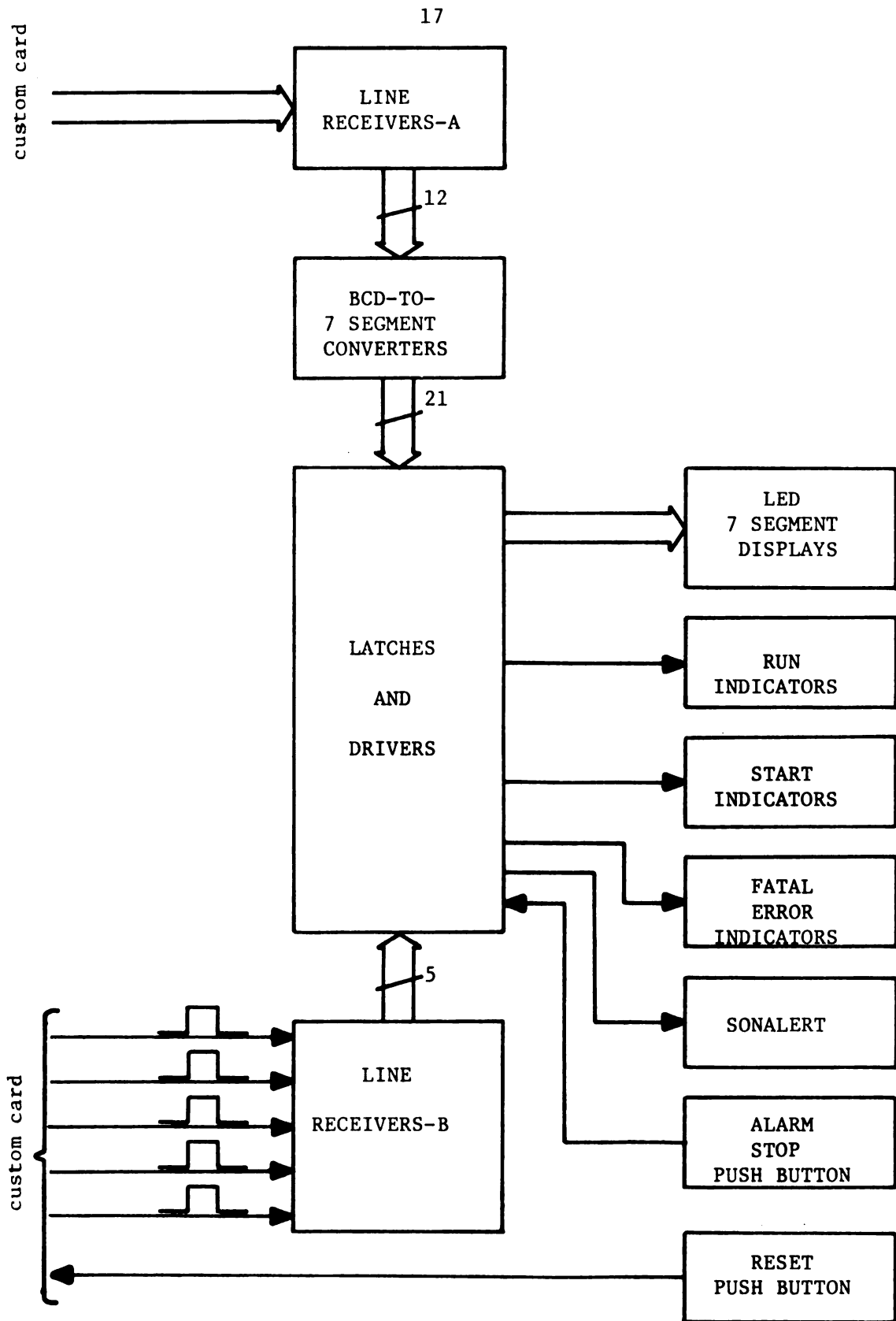


Figure 2.4 Display Controller Block Diagram

set turns off the audible alarm and the ALARM STOP lighted push button switch on the Display Panel; bit 1 controls the START indicator lights on the Local Teletype and the Display Panel; bit 2 controls the FATAL ERROR indicator lights on the Local Teletype and the Display Panel, and when set turns on the audible alarm and the flashing ALARM STOP lighted push button switch on the Display Panel; and bit 3 controls the RESET indicator light on the Display Panel.

OTA :15,4 - sets the seconds readout on the Display Panel from the A register. Bits 0 thru 3 must be set with the BCD code for the units digit, and bits 4 thru 7 must be set with the BCD code for the tens digit.

OTA :15,5 - sets the minutes readout on the Display Panel from the A register. Bits 0 thru 3 must be set with the BCD code for the units digit, and bits 4 thru 7 must be set with the BCD code for the tens digit.

OTA :15,6 - sets the hours readout on the Display Panel from the A register. Bits 0 thru 3 must be set with the BCD code for the units digit, and bits 4 thru 7 must be set with the BCD code for the tens digit.

OTA :15,7 - sets the days readout on the Display Panel from the A register. Bits 0 thru 3 must be set with the BCD code for the units digit, bits 4 thru 7 must be set with the BCD code for the tens digit, and bits 8 thru 11 must be set with the BCD code for the hundreds digit.

2.2.5 Local Modem/Controller

The Local Modem/Controller interfaces both the Local Teletype and the Remote Teletype to the Teletype port of the processor. The Custom Controller instructs the Local Modem/Controller to select which Teletype is active. When the Remote Teletype is selected, the modem translates the digital signal from the Teletype port into an audio frequency signal which is then sent to the Remote Modem/Controller via a dedicated telephone line.^(5,6) The remote modem then converts the received audio frequency signal into a digital signal which the Remote Teletype can accept. The use of modems and a dedicated telephone line allows the Remote Teletype to be located as far away as desired from the Central Station rack cabinet.

The Local Modem/Controller also generates the DCI control character when the Local REQUEST button is pushed. This character is sent to the Teletype port of the processor if the circuit was enabled by the Custom Controller.

The Local Modem/Controller turns the Local Teletype on and off when the Custom Controller sends it the proper signals. When the Local Teletype is first turned on, the Printer Enable Delay circuit insures that no data is sent to the Teletype printer for 600 msec. This delay allows the motor to come up to full speed first. A block diagram of the Local Modem/Controller is shown in Figure 2.5.

The Custom Controller decodes the following instructions which affect the operation of the Local Modem/Controller:

SEL :13,0 - turns off the carrier that the local modem transmits to the remote modem. The remote modem detects the absence of a carrier, turns off the RUN indicator light on the Remote Teletype, and then turns on the START indicator

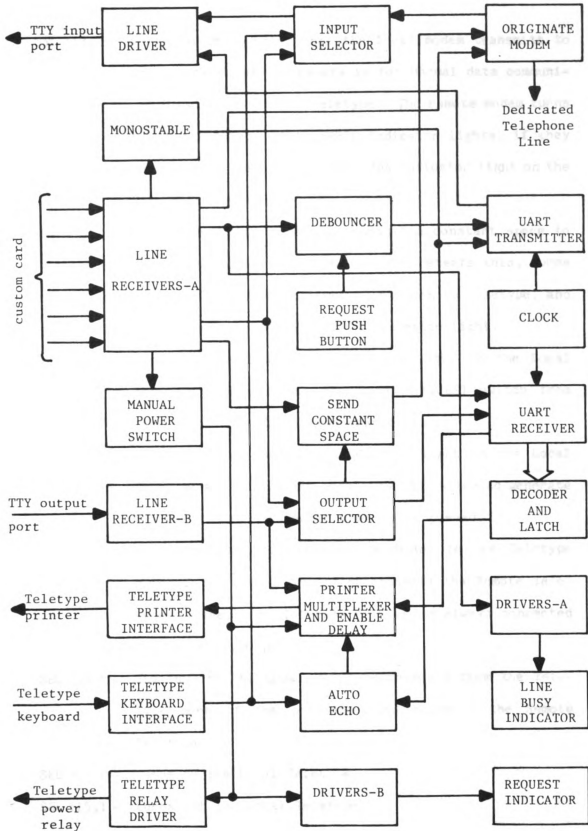


Figure 2.5 Local Modem/Controller Block Diagram

light.

- SEL :13,1 - turns on the carrier that the local modem transmits to the remote modem and resets it for normal data communication with the Remote Teletype. The remote modem turns off the START or FATAL ERROR indicator lights, if they were on, and then turns on the RUN indicator light on the Remote Teletype.
- SEL :13,2 - causes the local modem to transmit a constant space to the remote modem. The remote modem detects this, turns off the RUN indicator light on the Remote Teletype, and then turns on the FATAL ERROR indicator light.
- SEL :13,3 - turns on the LINE BUSY indicator light on the Local Teletype and prevents the local REQUEST switch from generating the DC1 control character.
- SEL :13,4 - turns off the LINE BUSY indicator light on the Local Teletype and allows the local REQUEST switch to generate the DC1 control character when it is pushed.
- SEL :13,5 - connects the Local Teletype keyboard to the Teletype port of the processor and disconnects the Remote Teletype. (The Local Teletype printer is always connected to the Teletype port.)
- SEL :13,6 - disconnects the Local Teletype keyboard from the Teletype port of the processor and connects the Remote Teletype.
- SEL :15,0 - turns on the Local Teletype.
- SEL :15,1 - turns off the Local Teletype.

2.2.6 Battery Real-Time Clock

The Battery Real-Time Clock is a battery-operated clock which continues to run during a power failure. When power is restored, the Real-Time Operating System reads the Battery Real-Time Clock to determine the current time. The DAS requires the current time to complete the restart procedure.

The Custom Controller reads the Battery Real-Time Clock and transfers the time to the minicomputer data bus when the correct sequence of instructions is executed. The Clock can also be set from the minicomputer data bus by the Custom Controller when another sequence of instructions is executed. A block diagram of the Battery Real-Time Clock is shown in Figure 2.6.

The Battery Real-Time Clock responds to the following instructions after they are decoded by the Custom Controller:

INA :15,0 - transfers the minutes and seconds time from the Battery Real-Time Clock to the A register. The seconds, in binary, are contained in bits 0 thru 5; the minutes, also in binary, are contained in bits 6 thru 11. The SEL :15,5 instruction must be executed first.

INA :15,1 - transfers the hours and days time from the Battery Real-Time Clock to the A register. The hours, in binary, are contained in bits 0 thru 4; the days, also in binary, are contained in bits 5 thru 13. The SEL :15,5 instruction must be executed first.

OTA :13,0 - sets a new time, from the A register, into the seconds and minutes counters in the Battery Real-Time Clock. The seconds counter is set from bits 0 thru 5; the minutes counter is set from bits 6 thru 11. The SEL

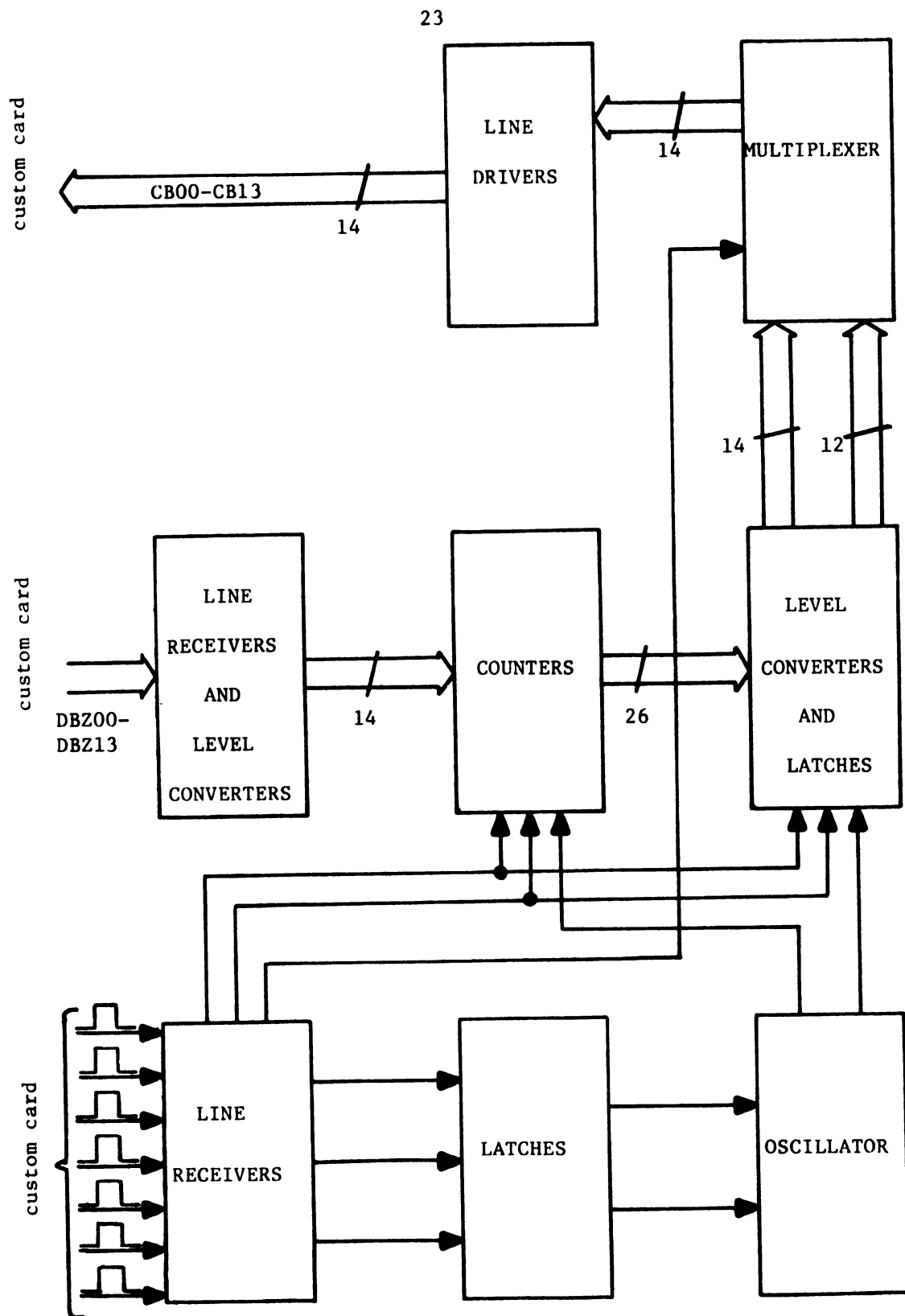


Figure 2.6 Battery Real-Time Clock Block Diagram

:15,6 instruction must be executed first.

OTA :13,1 - sets a new time, from the A register, into the hours and days counters in the Battery Real-Time Clock. The hours counter is set from bits 0 thru 4; the days counter is set from bits 5 thru 13. The SEL :15,6 instruction must be executed first.

SEL :15,5 - sets up the Battery Real-Time Clock so that it can be read. This instruction must be executed before either INA :15,0 or INA :15,1 is executed.

SEL :15,6 - enables the Battery Real-Time Clock to be set with a new time. This instruction must be executed before either OTA :13,0 or OTA :13,1 is executed.

SEL :15,7 - sets the Battery Real-Time Clock for normal operation. This instruction must be executed after the Clock is read or set with a new time.

2.2.7 Remote Modem/Controller

The Remote Modem/Controller interfaces the Remote Teletype to the dedicated telephone line. This dedicated telephone line connects the two Modem/Controllers and allows the Remote Teletype and the DAS to communicate with each other.

The Remote Modem/Controller generates the DC2 control character when the REQUEST button on the Remote Teletype is pushed. This control character is sent to the Local Modem/Controller over the dedicated telephone line. If the Remote Teletype is selected by the Local Modem/Controller, the control character is then sent to the Teletype port of the processor.

The Remote Modem/Controller turns the Remote Teletype on and

off; it contains the same delay circuit that the Local Modem/Controller does. Data is not presented to the Teletype printer until 600 msec after the Teletype is turned on. This allows the motor to come up to full speed first. A block diagram of the Remote Modem/Controller is shown in Figure 2.7.

The Remote Modem/Controller responds to the following ASCII control characters when they are received from the processor via the dedicated telephone line:

DC3 (:93) - turns on the Remote Teletype.

DC4 (:94) - turns off the Remote Teletype.

DLE (:90) - turns on the LINE BUSY indicator light on the Remote Teletype.

ETB (:97) - turns off the LINE BUSY indicator light.

NAK (:95) - enables auto-echo at the Remote Teletype: anything typed on the Teletype keyboard is immediately printed on the Teletype printer.

SYN (:96) - disables auto-echo.

2.3 Remote Station Organization⁽⁸⁾

A Remote Station consists of a rack cabinet and a single Data Coupler. A block diagram of a Remote Station is shown in Figure 2.8.

The modem is connected to a Bell System CBT Data Coupler. The Central Station begins the process of sampling a sensor by calling the Remote Station on a dial-up telephone line. The Data Coupler at the Remote Station detects the ringing signal on the telephone line and signals the modem. The modem then instructs the Data Coupler to take its telephone off-hook and establishes a data connection.

The Central Station next requests the Remote Station to sample a

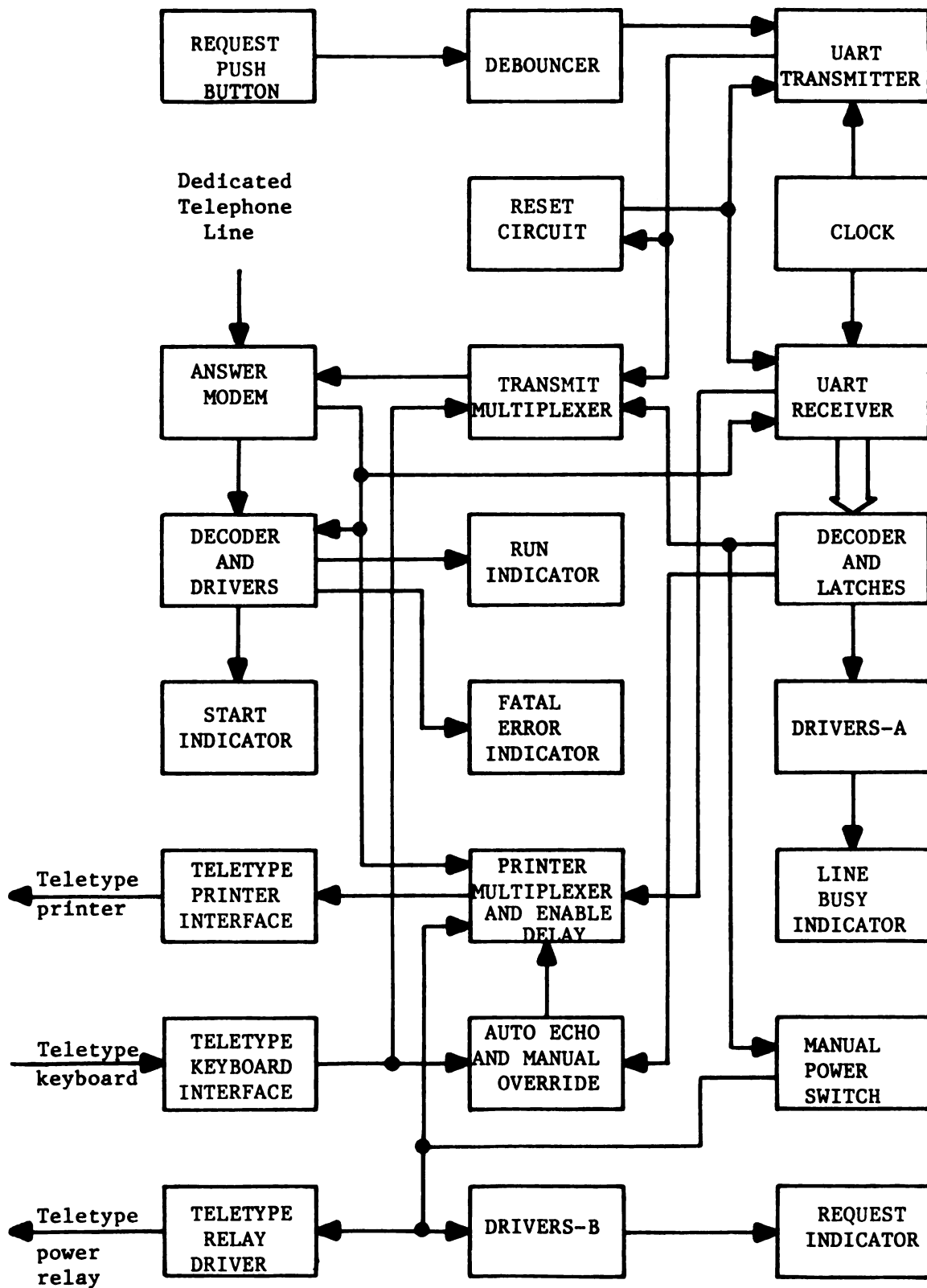


Figure 2.7 Remote Modem/Controller Block Diagram

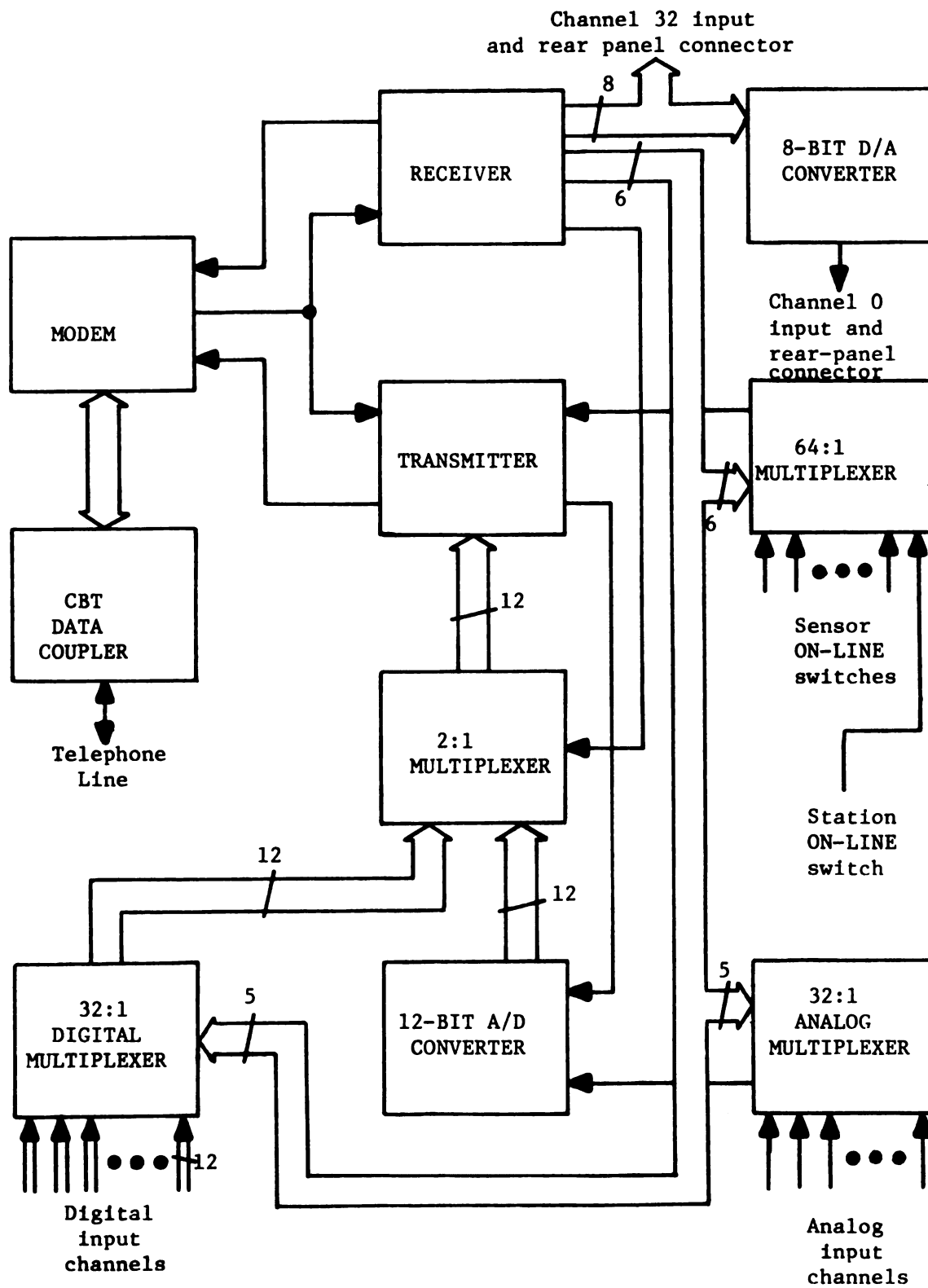


Figure 2.8 Remote Station Block Diagram

particular sensor by transmitting five ASCII characters to the Remote Station:

$$= C_1 C_2 \$?$$

The modem converts the received audio frequency tones from the telephone line into a digital signal which is sent to both the transmitter and receiver. The receiver decodes C_1 and C_2 , which are ASCII digits between 0 and 7. The lower three bits from these two characters are concatenated to form a six-bit address to select one of the 64 sensor input channels. Five of these bits select one of the 32 digital inputs and one of the 32 analog inputs. The most significant bit selects between the output of the digital multiplexer and the output of the analog to digital (A/D) converter. The six bits also select one of the 64 ON-LINE switches on the front panel of the Remote Station.

The transmitter then decodes the ? sent by the Central Station and commands the A/D converter to begin the conversion process. When the conversion is complete the transmitter begins transmitting five ASCII characters back to the Central Station. The first four ASCII characters contain the 12 bits of sensor data which are selected by the multiplexers. If the selected sensor's front panel switch is in the ON-LINE position, the four transmitted characters are ASCII digits between 0 and 7. The first character contains the three most-significant bits of the sensor data while the fourth character contains the least significant three bits. Otherwise, to indicate an off-line condition, bit 7 of each transmitted character is set, resulting in alphabetic characters being transmitted to the Central Station. The fifth character sent by the transmitter is a stop character; it must be sent for correct transmitter operation.

After the transmitter has completed sending the five characters to the Central Station, the Central Station may request the Remote Station to sample another sensor via the same sequence of characters. When the Central Station terminates the telephone connection, the modem at the Remote Station detects the loss of the carrier and puts its telephone on-hook. The Remote Station is now ready to accept another call.

The Remote Station includes a data path integrity check which allows a data pattern to be received from the Central Station and then transmitted back to the Central Station as a check on the operation of the Remote Station. This test feature is activated when the Remote Station receives the character string:

$$= P_1 P_2 C_1 C_2 \$?$$

C_1 and C_2 have the same function as explained above. The low-order 4 bits of P_1 and P_2 are concatenated to form an 8-bit test pattern, with the 4 bits from P_1 comprising the most significant half. The receiver decodes P_1 and P_2 and presents the 8-bit test pattern directly to channel 32. The test pattern also appears on the channel 32 connector on the rear panel of the Remote Station and is simultaneously converted to an analog voltage by a digital to analog (D/A) converter. This analog voltage is connected to channel 0 and also appears on the rear panel channel 0 connector. The ? causes the transmitter to encode the channel selected by C_1 and C_2 into five ASCII characters which are sent to the Central Station, as before. The Central Station can then compare the received and transmitted data pattern to determine if any errors occurred at the Remote Station. If C_1 and C_2 select a channel other than 0 or 32, a jumper cable must be connected between either the channel 0 or the channel

32 connector and the selected sensor input connector for the data path integrity check to function correctly.

CHAPTER III

SYSTEM OPERATION

3.1 Introduction

The DAS has three basic modes of operation: Automatic Mode, during which the sensors at the Remote Stations are sampled and the data stored at the Central Station; Dump Mode, during which the accumulated sensor data is transferred to the 6500 Computer system; and the Manual Mode, during which an operator is able to access the DAS via one of the Teletypes. Any two, or all three, of these basic modes may be occurring simultaneously.

In addition there are two error modes: The Disk Error Mode is entered if an error in one of the disk units occurs; the Power Restart Mode is entered when power is restored following a power failure. Both of these error modes lock out all other modes and do not allow normal DAS operation until they have finished.

Each of these five modes is explained in detail in this chapter.

3.2 Automatic Acquisition Mode

The Automatic Acquisition Mode is entered every fifteen minutes. The DAS checks the definitions for each sensor to determine which should be sampled. If any sensors at a particular Remote Station should be sampled, that station is called on a dial-up telephone line and a data connection is established. The Central Station then requests the Remote Station to sample each required

sensor and stores the received sensor data in the correct sensor data block on a floppy disk. The received data is also checked against the tolerance parameters for that sensor. An error message, indicating which sensors returned out-of-tolerance data, is printed on the Teletype(s) after the Automatic Acquisition Mode has completed sampling all the required sensors.

A transmission error may occur; the Central Station detects this and requests the Remote Station to retransmit the individual sensor's data. Three attempts are made to sample each sensor; if transmission errors occur all three times, a missing-data flag is written into that sensor's data block and the next required sensor at the Remote Station is sampled. If transmission errors prevent the Central Station from receiving data from three sensors at one Remote Station, the Central Station writes missing-data flags into the sensor data blocks for all remaining sensors that should be sampled at that Remote Station. The next Remote Station is then called if any sensors at it should be sampled. Missing-data flags are also written into a sensor data block if the sensor was placed off-line at either the Remote Station or the Central Station.

The telephone connection to a Remote Station may be accidentally broken due to equipment malfunction at the telephone company. In this instance the Central Station redials the Remote Station and attempts to reestablish a data connection. Missing-data flags are added to the appropriate sensors' data blocks if this attempt is unsuccessful.

When the Automatic Acquisition Mode has completed sampling the required sensors, it prints any accumulated error messages on the Teletype(s) as soon as they are available, since an operator may be

using one to access the DAS. Error messages are always printed on the Remote Teletype. They are also printed on the Local Teletype when the blue PRINT ERRORS push button switch on the Local Teletype is turned on. This switch locks in and lights up to indicate that it is turned on. Pushing the button a second time disables the local error listing. A typical error listing is shown below.

The time and date of the Automatic Acquisition is printed on the first line. The first number in each pair listed under the "OUT OF TOLERANCE" heading is a station number, and the second number is the sensor at that station which is out of tolerance. Data received from these sensors is still stored in the sensor data blocks. Sensors at the stations listed under the "NO REPLY" heading, which should have been sampled, have missing-data flags written into their sensor data blocks. If the Automatic Acquisition Mode cannot sample all the required sensors in 15 minutes, the message "NOT ENOUGH TIME TO SAMPLE ALL SENSORS" is printed and missing-data flags are written into the sensor data blocks of the sensors which could not be sampled.

14:30 6/30/77

THESE SENSORS ARE OUT OF TOLERANCE:

1,05 6,04 6,05 6,06

NO REPLY FROM THESE STATIONS:

8

NOT ENOUGH TIME TO SAMPLE ALL SENSORS

3.3 Dump Mode

This mode is entered every weekday morning after the 9:00 Automatic Acquisition Mode has finished. During the Dump Mode, a

24-hour block of data from each sensor is transferred to the 6500 Computer system. The 6500 is called on a dial-up telephone line, and a data connection is established. The DAS scans the 6500 header message and sends the password, problem number, and user ID at the correct time. After the 6500 has responded and completed logon procedures, the DAS requests the 6500 to begin executing the input processing program. This program prompts the DAS to send a condensed version of the sensor data line by line. The actual format used to transmit the sensor data is given in Chapter 4, section 4.8. Transmission accuracy is insured by including two checksum characters at the end of each line sent to the 6500. The 6500 recomputes the checksum for the line; if it doesn't match the checksum transmitted with the line, the 6500 requests the DAS to retransmit the same line. If any one line must be retransmitted five times, the DAS terminates the data connection to the 6500. The DAS then redials the 6500 and retransmits the same 24-hour block of sensor data from the beginning. When all the sensor data from one 24-hour block is transferred, the DAS requests the 6500 to add the sensor data to the data base. The DAS logs off the 6500 when this operation is completed. This procedure is then repeated if any more 24-hour blocks of sensor data must be sent to the 6500. More than one 24-hour block of sensor data is transferred if the 6500 was not available the previous day; this occurs on Monday mornings. Seven days of sensor data are automatically stored at the Central Station if the 6500 is continuously unavailable. If the 6500 is still not available on the eighth day, the oldest 24-hour block of sensor data is erased to store the sensor data from the past 24 hours.

3.4 Disk Error Mode

This error mode is entered if either floppy disk drive generates an error. When an error occurs, all the other modes that may have been active are immediately terminated. All normal DAS activities are suspended until the disk error is manually corrected. An error message, reproduced below, is printed on both Teletypes, and the red FATAL ERROR indicator lights on both Teletypes and the Display Panel of the Central Station are turned on. Additionally, a loud audible alarm is activated at the Central Station. This audible alarm may be silenced by momentarily pushing the flashing red ALARM STOP button on the Display Panel. After the cause of the disk error has been determined and corrected, pushing the blue RESET button on the Display Panel causes the DAS to check the operation of both disk drives. If they function correctly, the DAS enters the Power Restart Mode to begin restart processing. Appendix B contains a list of the error codes generated by the disk controller.

\$\$\$\$

\$\$\$\$ FATAL SYSTEM CRASH 15:34 6/30/77

\$\$\$\$ NOTIFY SYSTEM SUPERVISOR AS SOON AS POSSIBLE

\$\$\$\$

3.5 Power Restart Mode

The DAS detects when a power failure is about to occur and halts all processing. When power is restored, the DAS immediately enters the Power Restart Mode, but waits one minute before beginning restart processing in case another power failure occurs. After the minute has elapsed, the DAS reads the correct time from the Battery Real-Time Clock. Using this time, it adds missing-data flags to the

appropriate sensor data blocks to update them to the correct time. When restart processing is completed, the DAS prints the message below on both Teletypes and resumes normal processing. The DAS is able to correctly recover from a power failure that occurs while it is doing restart processing from a previous power failure. The yellow START indicator light on both Teletypes and the Display Panel of the Central Station indicates when the Power Restart Mode is active.

\$\$\$\$\$ RECOVERY COMPLETED 12:32 7/1/77

3.6 Manual Mode

The Manual Mode allows an operator to access the DAS via either of two Teletypes; only one Teletype may be in use at a given time. To request access, the blue REQUEST button on a Teletype is pushed. If the other Teletype is not in use, the DAS turns on the Teletype and prompts for a command by typing an asterisk (*). The operator may then type any one of the valid command letters. Typing an invalid command letter causes the DAS to prompt for a command again. The Teletype is automatically turned off if nothing is typed within 30 seconds. When one of the Teletypes is in use, the white LINE BUSY indicator light on the other Teletype is turned on; pushing the REQUEST button on that Teletype has no effect.

The current status of the DAS is displayed by three indicator lights on each Teletype. The green RUN light indicates that the DAS is functioning normally; the yellow START light shows that the DAS is in the Power Restart Mode; and the red FATAL ERROR light indicates that a disk error has occurred. The Manual Mode can be entered only when the green RUN light is on.

Two levels of security are provided to protect against unauthorized modifications of DAS parameters. The Level 1 password is requested if the operator attempts to place a sensor on-line or off-line at the Central Station, or if he attempts to change the Level 1 password itself. The Level 2 password is requested when new stations and sensors are defined, when existing stations and sensors are deleted, and when the Level 2 password itself is changed. Once the correct Level 2 password is entered, commands requiring the Level 1 password are automatically allowed. As soon as a correct password has been entered, any of the commands protected by that password may be executed without reentering the password, as long as the Manual Mode is not terminated. Once the DAS turns off the Teletype, the password(s) must be reentered during the next Manual Mode access.

Each of the valid commands is explained in detail in the following sections. The underscored characters are those typed by the operator.

3.6.1 A Command

This command places sensors on-line at the Central Station; data received from a sensor during the Automatic Acquisition Mode is then stored at the Central Station (if the sensor is also on-line at the Remote Station). If the sensor is off-line (see D command), a missing-data flag is stored in the sensor's data block whenever it is time to automatically sample the sensor.

The A command allows the letter "A" to be entered instead of a sensor number; this causes all sensors at the designated Remote Station to be placed on-line at the Central Station. Note that the back-arrow (←) is used to correct a typing error; sensor 27 at

station 8 is therefore placed on-line. The DAS checks that the requested station and sensor exist and prints an error message if a nonexistent station or sensor number is entered. This command is terminated by typing a period or escape (ESC) instead of a station number. Level 1 authorization is required to execute the A command.

```
#  A
STA= 1 SEN= 5
STA= 6 SEN= A
STA= 8 SEN= 24←7
STA= 9 SEN= 35 DOESN'T EXIST
STA= .
```

3.6.2 D Command

The D command places sensors off-line at the Central Station. When the Automatic Acquisition Mode samples a sensor which is off-line, a missing-data flag is stored in its sensor data block. The usage of the D command is identical to that of the A command. Level 1 authorization is likewise required before this command can be executed.

3.6.3 H Command

This command prints the sensor data which was automatically gathered by the Automatic Acquisition Mode during the previous 24 hours for a selected group of sensors. The data from a maximum of six sensors may be printed at the same time. The list of requested sensors is ended by following the last sensor number by a period rather than a space, or by entering a period and then a space for the following station number. The time and date that the listing begins is printed at the top of each page. A second page is automatically

printed if all the data cannot fit on a single page. The printing may be terminated at any time by typing (ESC); the current line finishes printing, and the DAS then prompts for another command. Note the missing-data flag for sensor 5 at station 1 at 7:00; it indicates that the sensor was off-line at that time or that invalid data was received.

```

*  H
    STA= 1 SEN= 5
    STA= 8 SEN= 27
    STA= .
    09:12    06/30/77
    TIME    WNSP-10M  TEMP-30M
           MILES/HR  DEG-FRHT
    09:00  +8.17E+00  +7.41E+01
    08:30  +7.35E+00
    08:15                      +7.18E+01
    08:00  +4.09E+00
    07:30  +5.54E+00  +7.04E+01
    07:00   -MD-
    06:45                      +6.89E+01
*

```

3.6.4 K Command

This command allows the Level 1 password to be changed. The new password must contain four alphanumeric characters. The DAS acknowledges the new password by printing "OK". Level 1 authorization is required before the command can be executed.

* K

TYPE LEVEL 1 PASSWORD:

TYPE NEW LEVEL 1 PASSWORD:

OK

3.6.5 P Command

This command prints the current value of any specified sensor. The Central Station calls the correct Remote Station, establishes a data connection, and requests the Remote Station sample the specified sensor. The scaled and labeled value of the physical parameter is then printed on the Teletype.

The error message "LINE 1 OCCUPIED" or "LINE 2 OCCUPIED" is printed if the telephone line is in use when the DAS expects it to be available. Both the Dump Mode and the Automatic Acquisition Mode may be using the telephone lines; in this case, the message "BOTH LINES IN USE" is printed. The P command may be attempted again in a few minutes. After either one of these two conditions occur, the DAS aborts the P command and prompts for a new command. If the Remote Station's telephone line is busy or the Remote Station doesn't answer, the message "STATION UNAVAILABLE" is printed. A data connection may be correctly established to the Remote Station, but no data is received from the requested sensor; the message "NO RESPONSE FROM SENSOR" is then printed.

A data connection remains established to a Remote Station until the operator requests the value of a sensor at another Remote Station or until the DAS automatically turns the Teletype off after 30

seconds of inactivity. Therefore,⁴¹ any sensor at the connected Remote Station may be sampled without waiting for the DAS to call the Remote Station again. A sensor may be sampled by the P command as often as desired. This command is terminated by typing a period or (ESC) instead of a station number.

* P

STA= 1 SEN= 5 BOTH LINES IN USE

* P

STA= 1 SEN= 5 WNSP-10M = +1.27E+01 MILES/HR

STA= 1 SEN= 5 WNSP-10M = +1.21E+01 MILES/HR

STA= 1 SEN= 34 DOESN'T EXIST

STA= 1 SEN= 7 NO RESPONSE FROM SENSOR

STA= 8 SEN= 27 LINE 2 OCCUPIED

* P

STA= 8 SEN= 27 TEMP-30M = +7.63E+01 DEG-FHRT

STA= 10 SEN= 15 STATION UNAVAILABLE

STA= .

*

3.6.6 S Command

This command prints a status report which lists the defined Remote Stations. If all the sensors at a Remote Station were off-line at the Remote Station during the most recent Automatic Acquisition Mode, a single asterisk is printed after the station number. Two asterisks are printed after a station number to indicate that all the sensors at that Remote Station were placed off-line at the Central Station (by the D command). The status of the individual sensors at any Remote Station may be determined by entering the

station number. The single and double asterisks printed after the sensor numbers have the same meaning as explained above. In the example below, sensors 3 and 8 at Remote Station 2 were placed off-line at the Remote Station, while sensor 4 was placed off-line at the Central Station by the D command.

```
* S
STATIONS = 1** 2      8*    15
STA= 1
SENSORS  = 5** 7** 8**  9**
STA= 2
SENSORS  = 1      2      3*    4**  5      6      7      8*
STA= .
*
```

3.6.7 T Command

This command prints the time and the date; the time is given in the 24 hour format. After the line finishes printing, the DAS prompts for a new command.

```
* T
09:46 07/01/77
*
```

3.6.8 U Command

This command changes the Level 2 password. The new password must consist of four alphanumeric characters. The DAS checks the new password and acknowledges it by printing "OK". In the example below, the first attempt to enter a new password included an illegal character and was rejected. The second attempt was successful. Level 2 authorization is required to execute this command.

* U

TYPE LEVEL 2 PASSWORD:

██████

TYPE NEW LEVEL 2 PASSWORD:

██████

INVALID PW

* U

TYPE NEW LEVEL 2 PASSWORD:

██████

OK

*

3.6.9 V Command

This command defines a new Remote Station. Each Remote Station must be given a unique identifying number between 1 and 16 inclusive. Up to 14 digits (including the letter "A", as explained below) may be entered for the Remote Station's telephone number. Typing "Y" in reply to "OK?" then adds the Remote Station. Typing "N" aborts the V command and the DAS then prompts for another command.

The DAS is able to dial thru more than one telephone exchange when calling a Remote Station. The letter "A" is typed in the telephone number at the point where a second dial tone is expected. In the example below, when the Central Station calls Remote Station 1, it first takes its telephone off-hook, waits for the first dial tone, dials a 9, waits for the second dial tone, and then dials the remainder of the number: 13138868449. The back-arrow is again used to correct an operator typing error. Level 2 authorization is required to execute this command.

V

1-STA= 1

2-PHONE= 9A1313888←68449

OK? Y ADDED

#

3.6.10 W Command

This command deletes an existing Remote Station. Typing a "Y" in response to "OK?" deletes the specified Remote Station, while typing an "N" aborts the W command. All of the sensors at a Remote Station are automatically deleted at the same time the Remote Station itself is deleted. Consequently, if the same Remote Station is later redefined (by the V command), each of the sensors at the Remote Station must also be redefined (by the X command).

Once a Remote Station has been deleted, all of the data stored from the sensors at that station can no longer be accessed by any Mode. Accordingly, all the sensor data that has not yet been transferred to the 6500 Computer system will be lost. The way to insure that sensor data is not lost is to first place the entire Remote Station off-line at the Central Station (using the D command), and then later remove the Remote Station with the W command, after the Dump Mode has completed the next data dump to the 6500. This will occur on the next weekday morning the 6500 is available.

The DAS aborts the W command if either the Automatic Acquisition Mode or the Dump Mode is processing data from the Remote Station that is to be deleted. In this case, the command may be attempted again in several minutes. The W command also aborts if an attempt is made to remove an undefined Remote Station. In the example below,

attempting to remove Remote Station 9 caused the DAS to abort W; after several minutes the W command was reexecuted, and this time Remote Station 9 was successfully deleted. Level 2 authorization is required to execute this command.

```
*  W
REMOVE STATION
1-STA= 9 ABORT
*  W
TYPE LEVEL 2 PASSWORD:
██████
REMOVE STATION
1-STA= 9
OK? Y DELETED
*
```

3.6.11 X Command

This command defines a new sensor. An example of the X command is given at the end of this section. After the sensor number (0 to 63 inclusive) is entered, the DAS checks if the sensor is already defined and aborts the X command if it is. The third entry, the sampling interval (SI), is a number between 1 and 96 inclusive. It gives the interval, in 15 minute units, between the times when the sensor is sampled by the Automatic Acquisition Mode. For example, with SI = 5, the sensor is automatically sampled every 75 minutes, with the first sample of each day occurring at midnight. The fourth, fifth, and sixth entries give the location on disk unit 0 where the sensor data block begins. The CYL entry must be between 11 and 26 inclusive, the SEC entry between 1 and 26 inclusive, and the WORD

entry between 0 and 63 inclusive. After the WORD entry is typed, the DAS checks that the new sensor data block does not overlap any sensor data block(s) already defined on the disk. If overlap would occur, the X command is aborted. For more information on placement of the sensor data blocks on the disk, consult Chapter 4, section 4.5. The seventh entry (UT) and the eighth entry (LT) specify the upper and lower tolerances, respectively, for the sensor. The data received from all sensors is in the form of an integer between 0 and 4095. If the data a sensor sends to the Central Station during the Automatic Acquisition Mode is equal to or greater than the upper tolerance, or equal to or less than the lower tolerance, an error message indicating an out-of-tolerance condition is printed on the Teletype(s). The data is still entered into the sensor data block, however. The ninth entry is the multiplier (MULT). The actual value of the physical parameter is computed by multiplying the received integer (0 to 4095) by the multiplier and then adding the offset (OFFS), entry ten, to the result. The multiplier and offset are entered in floating point notation; the acceptable formats are given in Figure 3.1. The eleventh entry is the LABEL that identifies the sensor when the H or P commands are executed. Any eight printing characters may be entered for the label. The twelfth entry is the UNITS, which also contains eight printing characters and is also printed by the H or P command.

After these 12 parameters are entered, the DAS asks if it should add the sensor by printing "OK?". Typing "Y" adds the sensor with the parameters as just entered. However, if an error in typing any of the parameters is now discovered, it can be corrected by typing the line number of the parameter; the DAS then requests that

parameter again. If it is desired instead to abort the command, typing "N" will do so. The escape key (ESC) may be struck at any step in the definition process and the command then also aborts. Level 2 authorization is required to execute this command.

* X

DEFINE SENSOR

1-STA= 1

2-SEN= 5

3-SI= 2

4-CYL= 11

5-SEC= 20

6-WORD= 30

7-UT= 4095

8-LT= 0

9-MULT= 2.5E-02

10-OFFS= 0.0E0

11-LABEL= WNSP-10M

12-UNITS= MILES/HH

OK? 10

10-OFFS= 1.0E0

OK? 12

12-UNITS= MILES/HR

OK? Y ADDED

*

$+1.25789E4$	$= 12,578.9$	⁴⁸	$9.4E-3$	$= 0.0094$
$1.25789E04$	$= 12,578.9$		$-9.400000E-3$	$= -0.0094$

The absolute value of all floating point numbers must be between $+1.701411E+38$ and $+1.701411E-38$.

Figure 3.1 Floating Point Format

3.6.12 Y Command

This command deletes a sensor. After the Remote Station number and sensor number are entered, the DAS verifies that the sensor does exist. If it does not, the Y command is aborted, as shown in the first example. The Y command also is aborted if either the Automatic Acquisition Mode or the Dump Mode is processing data from the same Remote Station that the sensor is located at. In this instance, the command may be reexecuted in a few minutes. Otherwise, typing "Y" in response to "OK?" deletes the desired sensor. Typing an "N" causes the Y command to abort, and the DAS then prompts for another command. Level 2 authorization is required to execute this command.

```
* Y
REMOVE SENSOR
1-STA= 2
2-SEN= 13 ABORT
* Y
REMOVE SENSOR
1-STA= 2
2-SEN= 3
OK? Y DELETED
*
```

CHAPTER IV

SOFTWARE ORGANIZATION

4.1 Introduction

The software is designed to implement a Real-Time Operating System. Each of the three basic modes--Automatic Acquisition Mode, Dump Mode, and Manual Mode--may be active simultaneously. This type of operation can be achieved because the speed of the minicomputer processor is several orders of magnitude faster than external events which interface to the minicomputer. For example, the Automatic Acquisition Mode may be processing data from a Remote Station when a character is received from the 6500 Computer system. The software first saves the machine status of the minicomputer before allowing the Dump Mode to input the character just received. When the Dump Mode finishes inputting the character and any other processing it must do, it restores the saved machine status. By saving and then restoring the machine status, the Automatic Acquisition Mode will continue executing at the same point it was at when the character from the 6500 was received. This ability to switch from one mode to another in response to external stimuli makes extensive use of the interrupt facility of the minicomputer.

4.2 Interrupts

An interrupt is a signal sent to the processor indicating that some external event has occurred. Typing a character on the Teletype generates an interrupt, as does receiving a block of characters from

a Remote Station. The software can instruct the processor to ignore certain interrupts and accept other interrupts. Once the processor is enabled to recognize a particular interrupt, it executes one instruction from a fixed location in core memory when that interrupt occurs. This instruction is usually a jump and store instruction to a routine. Typically, the routine first saves machine status so that the program which was interrupted can correctly resume when the interrupt routine finishes. The routine then does whatever processing is required, restores machine status, and returns to the interrupted program. The program resumes executing as if the interrupt had not occurred.

By restricting the amount of time that the interrupt routine is executing, it is possible to return to the interrupted program with only a small time loss. The Real-Time Operating System minimizes this time loss so that each of the three basic modes may be active simultaneously, with no loss of received data and no excessive time delays.

4.3 Station Definition Blocks

There may be a maximum of 16 Remote Stations; each one of these stations has its own station definition block in the core memory of the minicomputer. Eight consecutive words form each of these blocks; the blocks occur consecutively, starting with the Station 1 block. The beginning of this block is labeled with the name S1NUMB. The first word in each station definition block is an ID word. Bits 0 thru 7 of this word are set to :FF if the Remote Station isn't defined. If it is defined, these bits contain :00, and then bits 8 thru 15 contain the number of digits in the Remote Station's telephone number. The second thru eighth words of the station definition

block contain the digits of the telephone number; there may be a maximum of 14 digits, each represented in binary form. The first digit is in the upper byte (bits 8 thru 15) of the second word, the second digit is in the lower byte (bits 0 thru 7) of the second word, the third digit is in the upper byte of the third word, and so on.

4.4 Sensor Definition Blocks

Each of the 16 Remote Stations contains inputs for 62 sensors and two test channels. Each one of these possible inputs has two words reserved for it in core memory. The first word contains the disk address, which is the location on disk unit 0 where the sensor data block begins. Bits 0 thru 5 of this word are the word address (0 thru 63), bits 6 thru 10 the sector address (1 thru 26), and bits 11 thru 15 the cylinder address (11 thru 26). If the sensor isn't defined, this disk address word contains :FFFF. The second word is the sensor ID word. Bit 0 is set if the sensor is off-line at the Central Station (set by the operator using the D Command); bit 1 is set if the sensor was off-line at the Remote Station the last time the Automatic Acquisition Mode sampled it. Bits 8 thru 15 contain the sampling interval for the sensor, which indicates how often the Automatic Acquisition Mode will sample the sensor. This is a binary number between 1 and 96 and indicates, in 15 minute units, the time between samples. The disk address word of sensor 0 at station 1 is labeled SENOAD; the two words for each sensor follow consecutively by sensors at a station, and then consecutively station by station.

4.5 Floppy Disk Organization

Two floppy disks are utilized by the Central Station. Each disk is divided into 78 cylinders, labeled from 0 thru 77. Each cylinder

in turn is divided into 26 sectors, labeled from 1 thru 26. Each of these sectors contains 64 words, labeled from 0 thru 63. The smallest unit of information that may be read from or written to a disk is a sector.

Cylinders 1 thru 10 of both disks contain the same sensor definition blocks. There is one of these blocks for each of the 1024 possible sensor and test inputs. Each block contains 16 words; four blocks are stored per sector. The sensor definition blocks are arranged consecutively, with the definition block for sensor 0 at station 1 occupying the first 16 words of sector 1 of cylinder 1. The first two words of each sensor definition block contain the disk address of the sensor data block and the sensor ID word, respectively. The disk address word on the disk has the same format as the disk address word in core memory. The sensor ID word on the disk contains only the sampling interval in bits 8 thru 15. If a sensor is undefined, the entire sensor definition block for that sensor contains undefined data. The third and fourth words are the upper and lower tolerances, respectively. When the sensor is automatically sampled by the Automatic Acquisition Mode, any data that is outside of the tolerance parameters will cause an error message to print on the Teletype(s). The fourth and fifth words contain the multiplier, which is represented in floating point format. The sixth and seventh words contain the offset, also expressed in floating point format. The actual value of the physical parameter is computed by multiplying the received data by the multiplier and then adding the offset. The eighth thru eleventh words contain the eight ASCII characters comprising the label for the sensor, while the twelfth thru fifteenth words contain eight ASCII characters for the sensor's units.

The two disks are further divided into a total of eight sections. Section 1 consists of cylinders 11 thru 26 on disk unit 0. This section contains the current sensor data blocks for all defined sensors. An individual sensor data block contains one word of sensor data for each time the sensor was automatically sampled in the previous 24 hour period, plus one word which contains the time the sensor was last sampled by the Automatic Acquisition Mode. The sensor data block begins with the time word, followed by the most recent data word. The data words then follow consecutively, from the most recent to the oldest, which is the last word in the sensor data block. For example, if a sensor has a sampling interval of 2 (30 minutes), the sensor data block then contains 49 words. A sensor data block is automatically set up on each floppy disk when a new sensor is added via the X command. Sensor data blocks may be placed anywhere in section 1, but must not overlap. The X command does not allow a new sensor to be added if its sensor data block would overlap the blocks for sensors already defined.

4.6 Real-Time Clock

The minicomputer processor contains a real-time clock option which generates interrupts at fixed intervals of time. Connector J1 on the rear chassis of the minicomputer is jumpered to cause this interrupt to occur 120 times per second in synchronization with the power line frequency. The real-time clock interrupt causes the processor to execute the instruction at location :18, which is an increment and skip on zero instruction. Location :19 contains a negative number which is incremented each time the interrupt occurs. As long as the number in location :19 remains negative, only the

increment instruction is executed, and the interrupted program then immediately resumes executing. When the number in location :19 is incremented and the result is zero, the real-time clock generates a sync interrupt, which then calls RTCLK. RTCLK first saves machine status. It then stores the number -120 in location :19 so that the real-time clock sync interrupt will again call RTCLK in one second.

RTCLK stores the current DAS time in several variables: SEC, MIN, HOUR, DAY, MTH, and YEAR. Every time that RTCLK is called, it increments this stored time by one second. Once a day, at midnight, RTCLK calls BATUPD, which transfers the DAS time to the Battery Real-Time Clock. This insures that if a power failure occurs during the day the time of the crystal-controlled battery-operated clock will be extremely close to the actual time when power is restored.

Each time RTCLK is called it checks if PFSKIP is set. This flag variable is set if the Power Restart Mode is still doing restart processing. In this instance, RTCLK only updates the time displays on the Display Panel, by calling DISPLY, and then restores machine status and returns to the interrupted program. If PFSKIP isn't set, RTCLK checks PFSCAN. This flag variable is also set by the Power Restart Mode and indicates that the Automatic Acquisition Mode can resume normal operation when the DAS time reaches the Restart Time, which was set by the Power Restart Mode in variables RSMIN and RSHOUR. When the DAS time is the same as the time stored in these two variables, RTCLK resets PFSCAN and PFFLG.

The Automatic Acquisition Mode is enabled if this call to RTCLK causes the minutes variable, MIN, to increment to 0, 15, 30, or 45. RTCLK first checks DASFLG to see if the Automatic Acquisition Mode is already active. If so, RTCLK sets the terminate flag DASTRM, which

signals the Automatic Acquisition Mode to write missing-data flags into the sensor data blocks for any sensors which still have not been sampled. The Automatic Acquisition Mode then begins sampling the required sensors for the current sampling period. If the Automatic Acquisition Mode is not already active, RTCLK sets DASFLG, which enables the call to DASMGR from the PWRUP routine. At exactly 9:00, RTCLK also sets DUMPEN and DSKCEN; DUMPEN enables the Automatic Acquisition Mode to copy section 1 of the disk into another section for the Dump Mode, and DSKCEN enables the Automatic Acquisition Mode to begin the Dump Mode if it is a weekday. RTCLK next checks if the time is 20:00 or later. If it is, RTCLK resets DMPFLG, which causes DMPMGR to terminate the Dump Mode, if it is still active, the next time DMPMGR is called.

RTCLK next checks if any time-outs are enabled. When the Manual Mode expects to receive data from a Remote Station, it sets the negative of the maximum time, in seconds, it can wait for the data in variable MSMOUT; it then enables the time-out by setting MOFLG. RTCLK increments MSMOUT when MOFLG is set and then checks if MSMOUT is zero. If so, data from the Remote Station wasn't received within the time limit, and RTCLK calls P6. P6 stores the machine status saved by RTCLK, so that the program RTCLK interrupted may be correctly resumed when P6 finishes executing. In a similar fashion, the Dump Mode sets the negative of the maximum time it can wait for data from the 6500 Computer system in variable DDMOUT; it then enables the time-out by setting DOFLG. If RTCLK didn't call P6, it now increments DDMOUT, if DOFLG is set, and checks if DDMOUT is zero. If so, the Dump Mode didn't receive the data from the 6500 within the time limit, and RTCLK calls DTO to handle this error condition. DTO also

stores the machine status which was saved by RTCLK. The Manual Mode also sets a maximum time limit in variable TTMOU; it enables it by setting TOFLG when it is waiting for the operator to type something on one of the Teletypes. If RTCLK didn't call DTO, it increments TTMOU when TOFLG is set and checks if TTMOU is zero. If it is zero, the operator waited too long to type a character, and RTCLK calls OFF to terminate the Manual Mode access to the DAS. RTCLK next checks TMFLG. TMFLG is set by the Automatic Acquisition Mode when it has finished sampling all the sensors; it indicates that the error messages may be printed on the Teletypes. When TMFLG is set, and TTYFLG is reset, indicating that the Manual Mode isn't accessing the DAS, RTCLK calls E, which again transfers the machine status from RTCLK and supervises the printing of the error messages on the Teletype(s). If RTCLK didn't call one of these other routines, it restores machine status itself and returns to the interrupted program. A flow chart of the Real-Time Clock is shown in Figure 4.1.

4.7 Automatic Acquisition Mode

The Automatic Acquisition Mode is always entered from a loop in routine PWRUP. The flag variable DASFLG is continuously checked in the loop. Every quarter-hour RTCLK sets this variable. The next time thru the loop, PWRUP detects that DASFLG is set; PWRUP then calls DASMGR to supervise the Automatic Acquisition Mode. DASMGR first calls ERRINT, which writes the current time into the error buffer. Any errors detected during the Automatic Acquisition Mode will follow this time flag in the error buffer. DASMGR then checks each station ID word in core memory to determine if that Remote Station is defined. If a station is defined, TMSAMP is called.

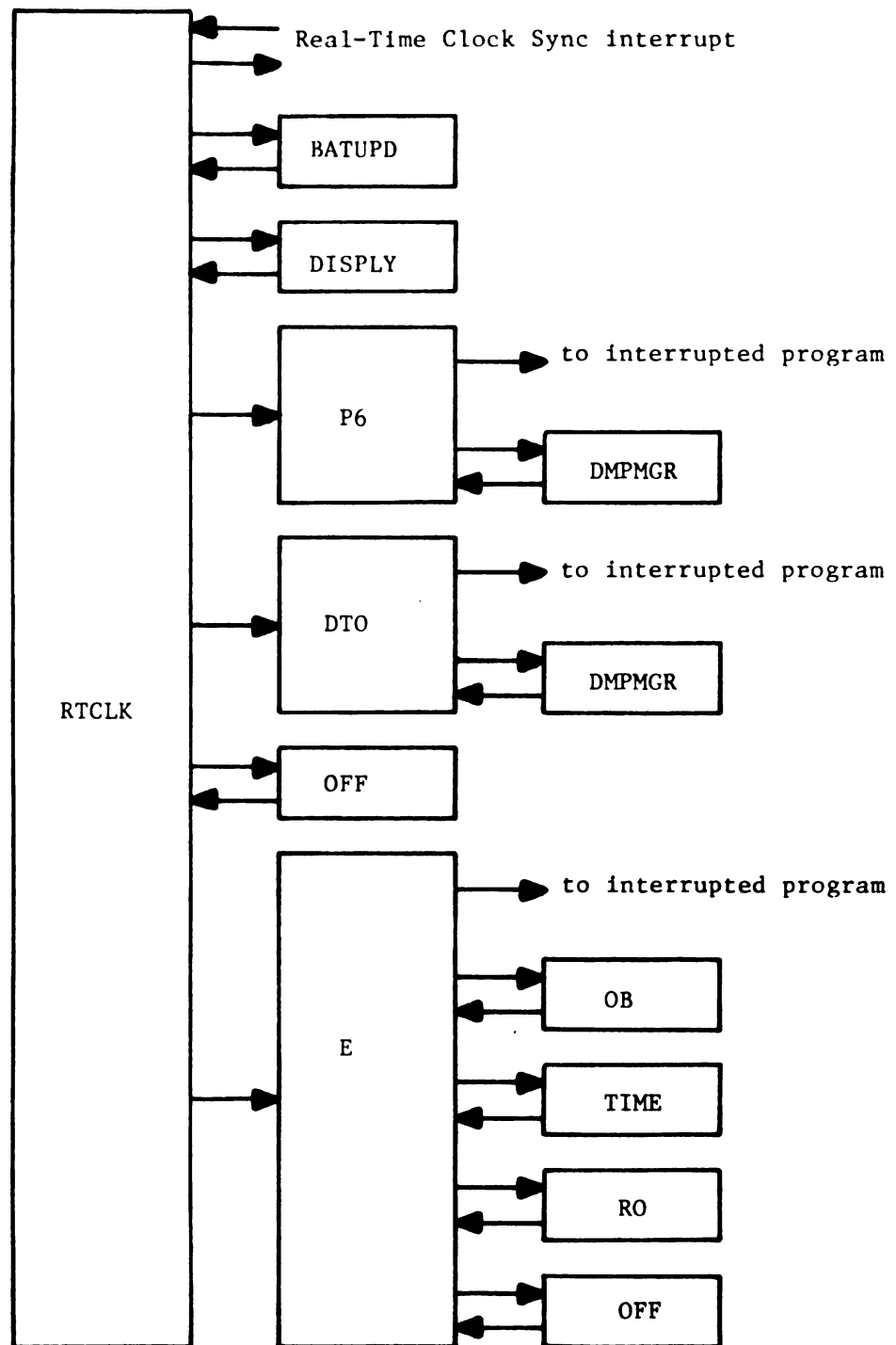


Figure 4.1 Real-Time Clock Flow Chart

TMSAMP checks all the defined sensors at the given Remote Station to determine if it is time to sample any of them. If none of the sensors are to be sampled, TMSAMP returns to DASMGR, which then checks the next station. TMSAMP also checks if all the sensors that must be sampled are off-line at the Central Station (bit 0 of the sensor ID word set). In this instance, TMSAMP returns to a different location in DASMGR, which then sets a flag variable, ABFL, so that missing-data flags will be entered into the correct sensor data blocks.

Once DASMGR determines that a Remote Station must be called, it checks the status of modem/calling unit #1. This modem/calling unit may be in use by the Manual Mode; DASMGR then sets a request flag, M1AREQ, and continuously checks the modem busy flag, M1BSY, until it is reset by the Manual Mode. DASMGR then resets its request flag, M1AREQ, and sets the addresses of the Automatic Acquisition Mode Asynchronous Modem Controller (AMC) and Automatic Calling Unit (ACU) interrupt routines at the correct base page locations. DASMGR next checks if the Manual Mode is connected, via modem/calling unit #2, to the same Remote Station that must be called. If it is connected, DASMGR sets a request flag, M2AREQ, to notify the Manual Mode to terminate its connection. DASMGR then continuously checks M2AREQ until it is reset by the Manual Mode when it disconnects the dialed station. DASMGR then dials the Remote Station by calling DIALA.

DIALA obtains the Remote Station's telephone number from the appropriate station definition block in core memory and sets up a block output instruction to send the telephone number to the ACU. DIALA issues a call request and then goes into a loop which checks if either DSS1FL or ACR1FL are set. If an Abandon Call and Retry (ACR) interrupt is generated by the ACU, DI5 is called. DI5 saves machine

status, sets ACRIFL, restores machine status, and returns to the interrupted program. When DIALA goes thru the loop again, it detects that ACRIFL is set and jumps out of the loop. DIALA then attempts to redial the Remote Station a second time. If the second attempt is also unsuccessful, DIALA returns to DASMGR. The Data Set Status (DSS) interrupt from the ACU calls routine DI6. DI6 saves machine status, sets DSSIFL, and stores the number of digits remaining to be dialed in CNITST. DI6 then restores machine status and returns to the interrupted program. The next time thru the loop DIALA detects that DSSIFL is set and jumps out of the loop. CNITST is zero if the call was successfully completed, and then DIALA waits 3 seconds for the telephone line to clear before returning to DASMGR. Otherwise DIALA redials the station a second time, as above.

DASMGR sets ABFL if the call to the Remote Station was unsuccessful. DASMGR then checks the sensor ID word in core memory for each sensor at the Remote Station to determine which sensors are defined. The sampling interval for each defined sensor is used to determine if it should be sampled. If it isn't time to sample the sensor, DASMGR checks the next sensor at that station. DASMGR checks ABFL if it is time to sample the sensor. If ABFL is set, or if the sensor is off-line at the Central Station (bit 0 of the sensor ID word in core memory set), a missing-data flag must be added to the sensor data block. DASMGR begins this process by calling DSKRD.

DSKRD begins executing and checks if another Mode is using the disk controller. If the disk controller is busy, DSKRD keeps checking the disk busy flag, DSKBSY, until the other Mode resets it. DSKRD then seizes control of the disk and reads the correct sensor data block from section 1 of the disk. The address of the sensor

data block is obtained from the sensor disk address word in core memory. After the read operation is completed, DSKRD returns to DASMGR, which then calls UPDATE.

UPDATE operates on the copy of the sensor data block in core memory. First the oldest data sample, obtained 24 hours previously, is deleted. Then the remaining data samples are moved down one word, and the new data sample, in this case a missing-data flag (:FFFF), is added to the top of the list. Finally UPDATE writes the time the Automatic Acquisition Mode began into the first word of the sensor data block and returns to DASMGR. DASMGR then calls DSKWR. The Automatic Acquisition Mode still has control of the disk controller since DSKRD didn't release it. DSKWR immediately writes the updated version of the sensor data block from core memory back to its correct location in section 1 of the disk. After the write operation is completed, DSKWR checks DSDREQ to find out if the Dump Mode has a request in for the disk controller. If DSDREQ is set, DSKWR executes a jump and store instruction to the address at location DSDADR. This transfers control to the Dump Mode. When the Dump Mode finishes with the disk, it returns to DSKWR. DSKWR then checks DSMREQ to see if the Manual Mode requested the disk. If DSMREQ is set, DSKWR executes a jump and store instruction to the address at location DSMADR. This transfers control to the Manual Mode. The Manual Mode returns to DSKWR when it finishes with the disk. DSKWR then returns to DASMGR. DASMGR calls LINE, which sets bit 1 of the sensor ID word in core memory to indicate that the sensor is off-line at the Remote Station. LINE returns, and DASMGR continues with the next defined sensor at the Remote Station.

When it is time to sample a sensor which is on-line at the

Central Station and ABFL is zero, DASMGR calls STD. STD begins executing and calls SAMPLE. SAMPLE first checks that the connection to the Remote Station was not broken. If it was, SAMPLE immediately returns to STD, which then returns to DASMGR. DASMGR calls DIALA to dial the same Remote Station again. However, the Remote Station is not called again if the connection was broken three times for the same Remote Station. In this case DASMGR sets ABFL, which causes missing-data flags to be written into the sensor data blocks for the remaining sensors that must be sampled at that station. If the Remote Station is still connected, SAMPLE sets up a block output instruction to send five characters to the Remote Station to request it to sample the desired sensors. A block input instruction is also set up to receive the five characters the Remote Station sends back. SAMPLE then enables AMC receive and transmit interrupts and enters a time-out loop. Two flag variables are checked in this loop: EXClFL and RLEOBF.

EXClFL is set in DI8, which is called when the AMC generates an exception interrupt. DI8 saves machine status, sets EXClFL, and inputs the AMC status word and saves it in AMClSW. DI8 then restores machine status and returns to the interrupted program. The exception interrupt is generated when a disconnect occurs or when an error in data transmission occurs. If an exception interrupt does occur, SAMPLE detects it when it checks EXClFL during the loop. SAMPLE then jumps out of the loop and checks the AMC status word. If AMClSW indicates a disconnect occurred, SAMPLE returns to STD, as before. If only a data transmission error occurred, SAMPLE does a 1.5 second time-out and then repeats the transmission to the Remote Station. If the received data contains errors again, SAMPLE returns to STD, which

calls LINE to set the sensor off-line at the Remote Station (bit 1 of the sensor ID word set). STD then calls DSKRD, which reads the sensor data block from section 1 of the disk. DSKRD completes the read operation and returns to STD, which calls DSKWR to write the updated sensor data block from core memory back to its correct location in section 1 of the disk. STD then returns to DASMGR.

RLEOBF is the other variable checked in the time-out loop. It is set in DI7, which is called when the AMC generates a receive end-of-block interrupt. DI7 saves machine status, sets RLEOBF, then restores machine status, and returns to the interrupted program. SAMPLE checks RLEOBF each time it goes thru the time-out loop and jumps out of the loop when RLEOBF is set. SAMPLE then determines if the sensor is off-line at the Remote Station, which will be the case if bit 6 of each character received is set. In this case SAMPLE returns to STD. Otherwise SAMPLE first reformats the five characters received from the Remote Station into a single word containing the 12 bits of data from the sensor and then returns to STD. STD updates the off-line bits in the sensor ID word; bit 1 is set if the sensor is off-line and reset if the sensor is on-line. STD uses the same procedure explained above to add either a missing-data flag or the sensor data word to the sensor data block. However, if the sensor is on-line, STD checks the returned data against the sensor tolerance parameters which DSKRD reads from the sensor definition block on the disk. If the data is less than or equal to the lower tolerance, or greater than or equal to the upper tolerance, STD calls ERROR. ERROR writes the station and sensor number and the code for the out-of-tolerance condition into the error buffer. ERROR then returns to STD, which returns to DASMGR.

SAMPLE may repeat the time-out loop without EXClFL or RLEOBF being set. This occurs if the Remote Station doesn't send the five characters fast enough. If neither flag variable is set within 2 seconds, SAMPLE exits the loop, resets the AMC interrupts, and then repeats the transmission to the Remote Station. If the time-out also elapses the second time, SAMPLE returns to STD, which writes a missing-data flag into the sensor data block, as explained above.

DASMGR keeps track of how many sensors at a station could not be sampled because of data transmission errors. If three sensors at a given station cannot be sampled for this reason, DASMGR sets ABFL and calls ERROR to write the station number and this error code number into the error buffer. When ERROR returns, DASMGR then writes missing-data flags into the sensor data blocks of any remaining sensors at that station. After all the sensors at a Remote Station have been processed, the connection to that Remote Station is terminated. DASMGR then resets ABFL and other flags and repeats this sequence of events for the next Remote Station.

DASMGR checks if the flag variable DASTRM is set after it samples each sensor. This variable is set by RTCLK if the Automatic Acquisition Mode has been active for longer than 15 minutes. When DASTRM is set, DASMGR writes missing-data flags into the sensor data blocks for all the sensors that must still be sampled. This insures that the next automatic acquisition can begin as soon as possible.

After all the required sensors at all Remote Stations are sampled, DASMGR releases modem/calling unit #1. It then checks if DSKCEN is set. This flag variable is set by RTCLK at 9:00 every morning. When DSKCEN is set, DASMGR copies the entire section 1 of the disk, which contains the current sensor data blocks, to one of

the other section on the disks. This copied version of the sensor data blocks is then used by the Dump Mode to transfer the sensor data to the 6500 Computer system. DASMGR searches for the first available section (#2 thru #8) and then calls DSKCPY to copy section 1 into that section. If all of the sections are filled, DASMGR instructs DSKCPY to overwrite the section containing the oldest sensor data blocks. DSKCPY interacts with the disk controller via the same sequence of instructions that DSKRD uses. When DSKCPY returns, DASMGR checks the dump enable flag, DUMPEN, which is also set by RTCLK at 9:00 every morning. If DUMPEN is set, and it is a weekday, DASMGR sets the dump flag, DMPFLG, and calls DMPMGR. DMPMGR begins executing; it starts the Dump Mode processing and then returns to DASMGR. DASMGR next sets TMFLG to enable the printing of any error messages accumulated during the Automatic Acquisition Mode, resets DASFLG, DSKCEN, and other flags, and returns to the loop in PWRUP to wait for RTCLK to set DASFLG again. A flow chart of the Automatic Acquisition Mode is shown in Figure 4.2.

4.7.1 Error Message Print Routine

E prints the error messages, which were generated during the Automatic Acquisition Mode, on the Remote Teletype and optionally on the Local Teletype. E executes as a Manual Mode routine, in that it prints on the Teletypes, but it is initially called from RTCLK. When the Automatic Acquisition Mode finishes processing, it sets TMFLG to indicate that the error buffer is complete. RTCLK checks TMFLG when it is called every second; when TMFLG is set and TTYFLG is reset, indicating that the Manual Mode isn't accessing the DAS, RTCLK calls E.

E begins executing and first transfers the machine status which

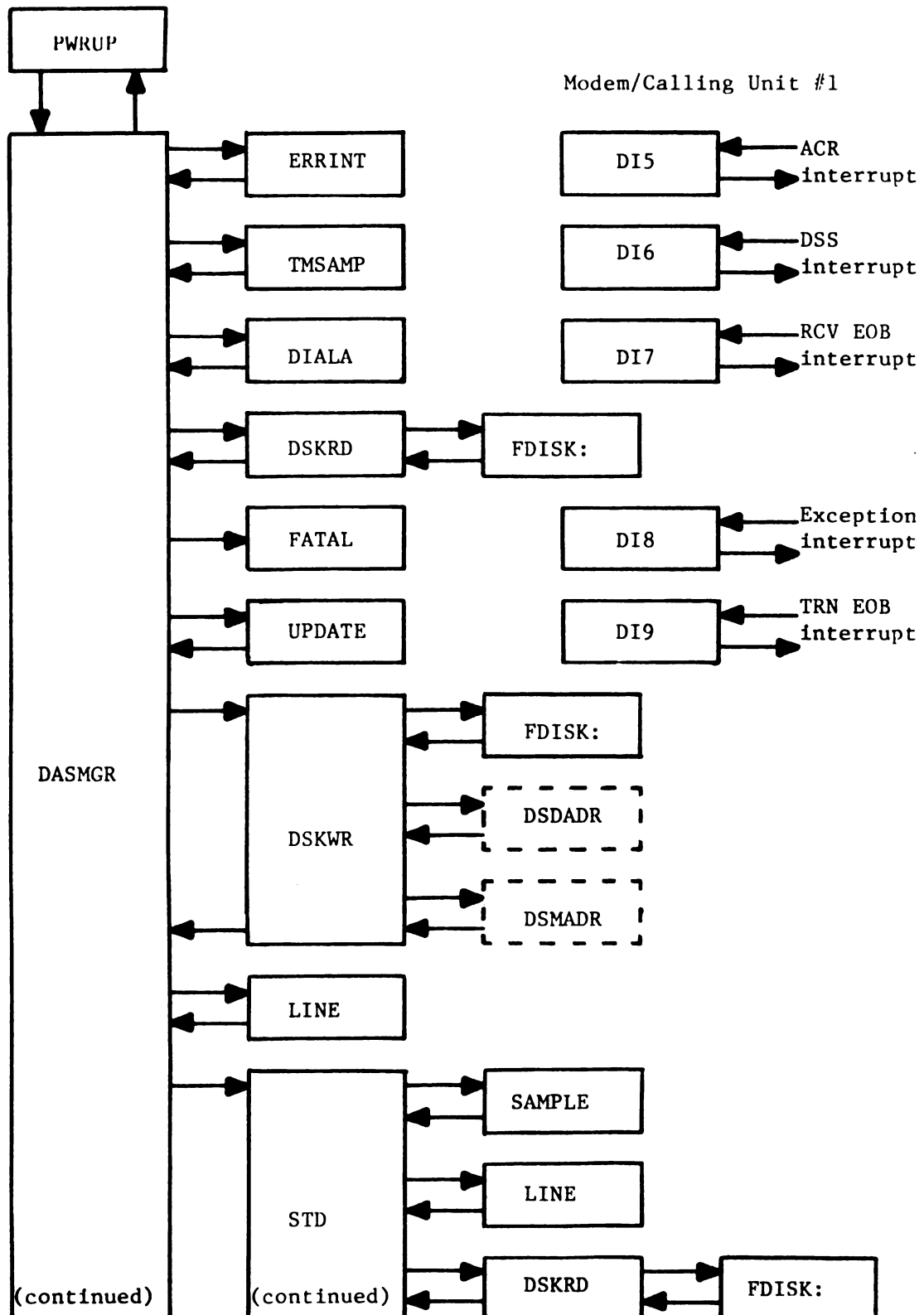


Figure 4.2 Automatic Acquisition Mode Flow Chart

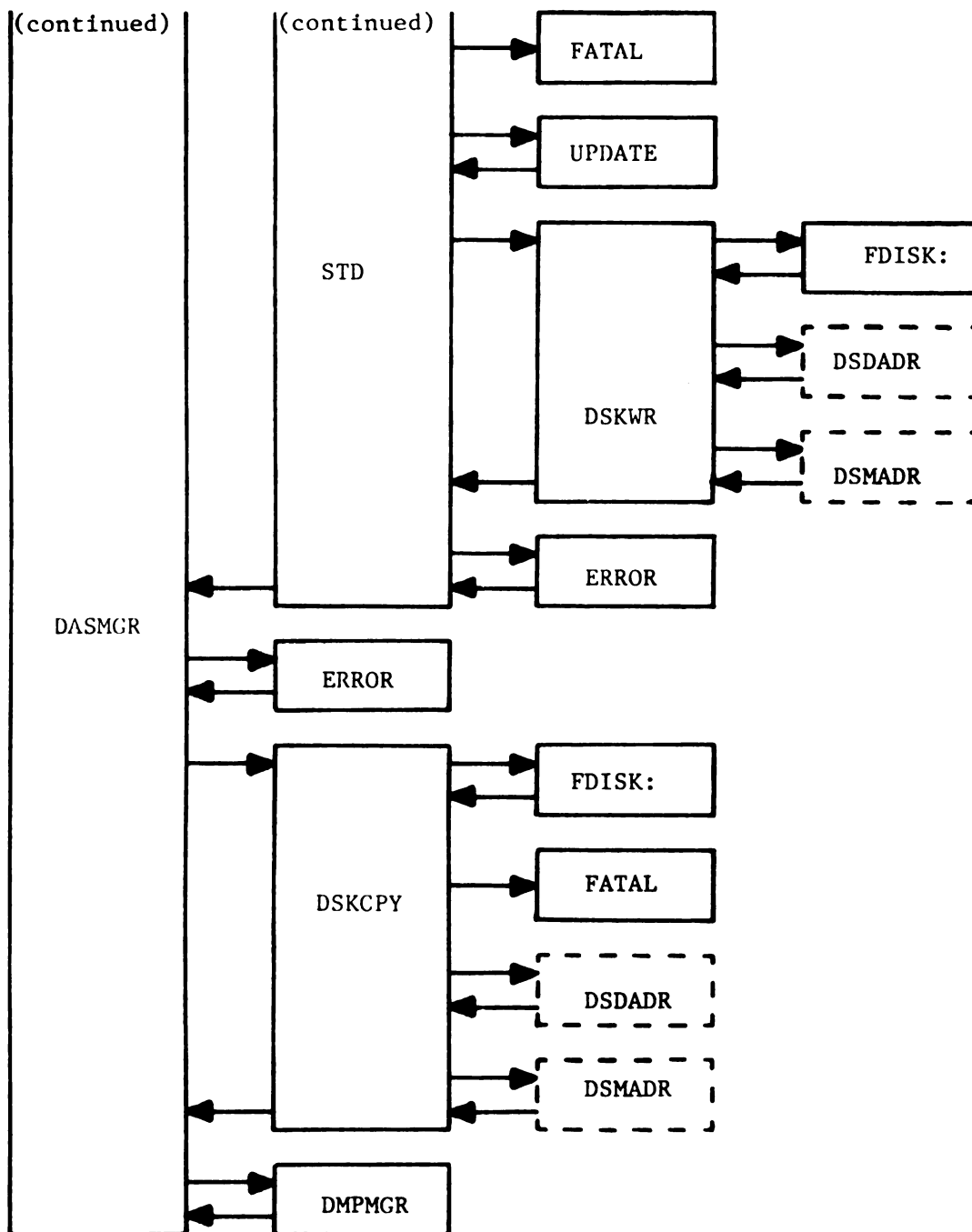


Figure 4.2 (cont'd)

was saved by RTCLK. E then resets the TTY input interrupts so that the Manual Mode cannot gain access to the DAS. E also writes the address of a second entry point, E7, into location MPOUT. E next checks the error buffer to determine if any error messages must be printed. If the error buffer doesn't contain any error codes, E enables TTY input interrupts; subsequently pushing one of the Teletype REQUEST push buttons will cause a TTY input end-of-block interrupt and initiate Manual Mode access. E then restores the transferred machine status and returns to the interrupted program.

When the error buffer does contain error codes, E enables output to the Remote Teletype (SEL :13,6). E then checks if the error messages should also be printed on the Local Teletype by sensing if the PRINT ERRORS push button switch on the Local Teletype is activated (SEN :13,0). If it is, the Local Teletype is turned on (SEL :15,0). E next obtains the time that the Automatic Acquisition Mode began. This time is written into two words in the error buffer at the start of the Automatic Acquisition Mode by ERRINT. E keeps track of what location it should check in the error buffer by the variable ERRD.

The first block of output sent to the Teletype(s) contains the control codes to turn the Remote Teletype on (:93) and disable auto-echo at both Teletypes (:96). This block also includes a line containing the time and date that the Automatic Acquisition Mode started processing. E then sets up the block output instruction, enables TTY output interrupts, restores the transferred machine status, and returns to the interrupted program.

After the line finishes printing, the TTY output end-of-block interrupt calls TTYOUT, which then calls E thru the E7 entry point.

E checks the error buffer for any out-of-tolerance error codes. A single word is used to represent each sensor which was out of tolerance during the Automatic Acquisition Mode sampling. Bits 0 thru 5 of each error code word contain the sensor number, bits 6 thru 10 contain the station number, bit 12 is set, and bits 13 thru 15 are zero. The error code for the out-of-tolerance condition is indicated by this pattern in bits 12 thru 15. The error buffer might not contain any out-of-tolerance error codes; E then checks the next type of error code. However, if E does find any of these codes in the error buffer, it sets up a block output instruction to print the message "THESE SENSORS ARE OUT OF TOLERANCE:". E then enables TTY output interrupts and returns thru TTYOUT to the interrupted program. After the message finishes printing, E is called from TTYOUT and locates each out-of-tolerance error code in the error buffer. The station and sensor numbers are converted into ASCII characters which can be printed. An output buffer is then set up which contains the ASCII characters for the station and sensor numbers. A maximum of ten station and sensor numbers are added to the output buffer. E then sets up a block output instruction, enables TTY output interrupts, and returns to TTYOUT, which restores machine status and returns to the interrupted program.

When the line finishes printing, the TTY output end-of-block interrupt calls TTYOUT, which calls E thru the E7 entry point. E then checks if any more out-of-tolerance error codes remain in the error buffer. If there are more, it formats another line of output as explained above. Otherwise E checks if there are any error codes in the buffer for Remote Stations which didn't respond to the Automatic Acquisition Mode request to sample a sensor. The station

number is again contained in bits 6 thru 10 of the error code word, but now bits 12 and 13 are set and bits 14 and 15 are zero. If any of these error codes are found in the error buffer, E sets up a block output instruction to print the message "NO REPLY FROM THESE STATIONS:". TTY output interrupts are then enabled, and E returns to TTYOUT, which returns to the interrupted program. When this message finishes printing, TTYOUT calls E thru the E7 entry point. E then searches the error buffer for all the error codes for stations which didn't reply. E next sets up an output buffer which contains the ASCII codes for these station numbers. E then sets up the block output instruction, enables TTY output interrupts, and returns to TTYOUT, which returns to the interrupted program.

When E is called again from TTYOUT, it checks if the error buffer contains the code word which the Automatic Acquisition Mode writes if it doesn't have enough time to sample all the sensors before the next sampling period must start. The error code word for this condition has bits 12, 14, and 15 zero and bit 13 set. If E finds this code, it sets up a block output instruction to print the message "NOT ENOUGH TIME TO SAMPLE ALL SENSORS". E then enables TTY output interrupts and returns to TTYOUT, which returns to the interrupted program.

After this line finishes printing, TTYOUT calls E thru the E7 entry point. E first erases all of the error codes it just processed from the error buffer and then updates ERRD. E next calls OFF to turn both Teletypes off and reset the LINE BUSY indicator lights. OFF returns to E, which returns to TTYOUT, which then returns to the interrupted program. After OFF finishes executing, the Manual Mode may again access the DAS.

4.8 Dump Mode

The Dump Mode begins when DASMGR calls DMPMGR at 9:00 every weekday morning. DMPMGR first checks the modem/calling unit #2 busy flag, M2BSY. If modem/calling unit #2 is being used by the Manual Mode, and the idle flag, M2IDLE, isn't set, DMPMGR sets its request flag, M2DREQ, and returns to DASMGR. When the Manual Mode finishes with modem/calling unit #2, it calls DMPMGR, which then proceeds. If the idle flag is set, the Dump Mode immediately seizes modem/calling unit #2. DMPMGR first initializes it and sets the addresses of the Dump Mode Asynchronous Modem Controller (AMC) and Automatic Calling Unit (ACU) interrupt routines at the correct base page locations. DMPMGR then checks which section of the disk is to be transferred to the 6500 Computer system and saves the section number. DMPMGR next checks the status of the telephone line. If it is occupied (i.e. the handset is off-hook), DMPMGR sets a 10 second time-out in RTCLK and returns. When 10 seconds have elapsed, RTCLK calls DTO, which transfers the machine status already saved by RTCLK. DTO resets modem/calling unit #2 and then calls DMPMGR, which repeats the check of the telephone line. When DMPMGR returns to DTO, machine status is restored, and the interrupted program resumes executing.

Once the telephone line is clear, DMPMGR sets up a block output instruction, which the ACU uses to dial up the 6500 Computer system. DMPMGR enables the ACU interrupts and returns. The ACU will then generate either an Abandon Call and Retry (ACR) interrupt or a Data Set Status (DSS) interrupt. The ACR interrupt is initiated if the call is not completed within approximately 40 seconds. The ACR interrupt calls DACU3, which first saves machine status. DACU3 then resets ACU #2, enables a 5 second time-out in RTCLK, and restores

machine status. The interrupted program then resumes executing. When five seconds have elapsed, RTCLK calls DTO, which then calls DMPMGR, as before. The 6500 is then redialed via the same procedure.

The ACU may also generate a DSS interrupt. In this case, DACU is called and saves machine status. DACU then checks if all the digits in the phone number were dialed. If they weren't, the call was unsuccessful and DACU sets a 5 second time-out in RTCLK, as was done for the ACR interrupt. DACU then restores machine status and returns to the interrupted program. If all the digits were dialed, DACU sets up a block output instruction, which generates an end-of-block interrupt after four control characters are sent to the 6500. The four control characters inform the 6500 that data transmission will be at 300 baud. DACU also resets the exception interrupt enable flag variable, DEXCEN, so that any exception interrupts generated will be ignored. DACU then writes the address of routine DUMPl into location DRCV, sets a 2 minute time-out in RTCLK, enables AMC transmit interrupts, restores machine status, and returns to the interrupted program.

After the four control characters are sent to the 6500, the AMC transmit end-of-block interrupt calls DAMC1. DAMC1 saves machine status, enables the AMC receive interrupts, restores machine status, and returns to the interrupted program. When the first character is received from the 6500, the AMC receive end-of-block interrupt calls DAMC. DAMC first saves machine status and then calls the routine whose address is stored in location DRCV. DUMPl therefore begins executing. It first sets DEXCEN so that the AMC status word is checked whenever an exception interrupt occurs. It then checks if the character just received from the 6500 is the last one in the

standard 6500 sign-on message. If there are still more characters, DUMP1 sets up a block input instruction to generate an end-of-block interrupt when another character is received from the 6500. DUMP1 then returns to DAMC, which restores machine status and returns to the interrupted program. When DUMP1 is called and detects the last character of the sign-on message, it disables the 2 minute time-out which was set in RTCLK and sets up a block output instruction to send the password, problem number, and user ID to the 6500. DUMP1 also sets up a block input instruction to receive one character from the 6500. The address of routine DUMP2 is then written into location DRCV. DUMP1 sets and enables a 3 minute time-out in RTCLK, enables AMC transmit interrupts, and returns to DAMC, which restores machine status and returns to the interrupted program.

After the password, problem number, and user ID are sent to the 6500, the AMC generates a transmit end-of-block interrupt. DAMC1 is called; it enables AMC receive interrupts and returns to the interrupted program. When the 6500 sends a single character, the AMC generates a receive end-of-block interrupt which calls DAMC. DAMC saves machine status and then calls DUMP2. DUMP2 immediately calls DREADY, which scans each character received from the 6500 to determine when the standard "READY" message has been received. If all the characters of the "READY" message have not been received, DREADY sets up a block input instruction to receive another single character from the 6500 and returns to DUMP2, which returns to the interrupted program thru DAMC. When DREADY is called and detects that all the characters of the "READY" message, except the last one, have been received, it returns to a different location in DUMP2, which then sets the address of a second entry point, DUMP2B, into location DRCV.

DUMP2 returns to the interrupted program thru DAMC. When the next character is received, the AMC receive end-of-block interrupt calls DAMC, which then calls DUMP2B. DUMP2B resets the time-out and sets up a block output instruction to send a command to the 6500. This command instructs the 6500 to begin executing the program which reads the characters sent by the Dump Mode. DUMP2B then sets the address of DUMP3 at location DRCV and again sets up a block input instruction to generate an end-of-block interrupt when one character is received. DUMP2B then enables a 3 minute time-out in RTCLK, enables AMC transmit interrupts, and returns to DAMC, which restores machine status and returns to the interrupted program.

The AMC transmit end-of-block interrupt again calls DAMC1, which enables AMC receive interrupts and returns to the interrupted program. When the first character is received from the 6500, the AMC receive end-of-block interrupt calls DAMC, which calls DUMP3. DUMP3 checks the character to see if it is a ":" (:3A) or a "?" (:3F). If neither of these characters was received, DUMP3 sets up a block input instruction to receive another single character from the 6500 and then returns thru DAMC to the interrupted program. DUMP3 continues to check each character received until it encounters a ":" or "?", or until the 3 minute time-out occurs. The 6500 inputting program sends back a ":" when it begins executing. DUMP3 is called by DAMC, detects the ":", and resets the time-out in RTCLK. It then calls DDISK to read the section header from the disk. DDISK begins executing and checks the disk busy flag, DSKBSY. If another Mode is using the disk controller, DDISK sets a second entry address at location DSDADR and sets its request flag, DSDREQ. It then returns to DUMP3, which returns thru DAMC to the interrupted program. When

the other Mode is finished with the disk, it checks the Dump Mode request flag. Since it is set, the other Mode executes a jump and store instruction to the address in location DSDADR, which then transfers control back to DDISK. DDISK writes the return address in the first location of DUMP3 so that it will correctly return to the other Mode. DDISK then reads the section header from the disk and sets up a buffer containing the ASCII characters for the month, day, and year that section 1 was copied. When this is completed, DDISK checks if the Manual Mode has requested the disk. If the Manual Mode disk request flag, DSMREQ, is set, DDISK executes a jump and store instruction to the address in location DSMADR. This transfers control to the Manual Mode, which then accesses the disk. When the Manual Mode is finished with the disk, it returns to DDISK, which returns to DUMP3.

DUMP3 next sequentially checks each station ID word in core memory to determine which stations exist. If a station does exist, DUMP3 sequentially checks the sensor ID words in core memory to determine which sensors at that station are defined. When DUMP3 locates a defined sensor, it calls DDISK to read the sensor data block off the correct section of the disk. DDISK gains access to the disk controller by the procedure explained above. DDISK also reads the sensor definition block from the disk to obtain the multiplier and offset for the sensor. It then converts the multiplier and offset into a string of twelve ASCII characters and returns to DUMP3. DUMP3 sets up an output block instruction to send 81 characters to the 6500. The eight ASCII characters comprising the section header are placed in the output buffer first, followed by two ASCII characters for a "a" (:E1). These two characters mark the beginning

of each sensor's data as transmitted to the 6500. The next two ASCII characters added to the output buffer represent the station number and the sensor number. These are followed by the twelve characters into which the sensor's multiplier and offset are coded. The next two characters contain the sampling interval for the sensor. Each data point in the sensor data block is then converted into two ASCII characters, which are placed consecutively into the output buffer, until it contains 78 ASCII characters. DUMP3 then calls CKSUM, which computes a checksum based on the 78 ASCII characters in the output buffer. CKSUM converts the checksum into two ASCII characters, places them into the output buffer, and then adds a carriage return to the end of the buffer. CKSUM returns to DUMP3, which sets up a block output instruction to send all 81 characters to the 6500 and sets up a block input instruction to receive one character from the 6500. DUMP3 then sets and enables a 3 minute time-out in RTCLK, enables AMC transmit interrupts, and returns. The interrupted program then resumes executing.

After all 81 characters are transmitted, the AMC transmit end-of-block interrupt calls DAMC1, which enables AMC receive interrupts and then returns. The 6500 inputting program receives the line of 81 ASCII characters and computes the checksum for the first 78 characters. If the newly computed checksum doesn't match the checksum contained in the 79th and 80th received characters, a transmission error occurred. The 6500 inputting program then sends a "?" to the Dump Mode. DAMC calls DUMP3 every time a single character is received from the 6500. When DUMP3 is called and detects the "?", it sets up a block output instruction to retransmit the same line of 81 ASCII characters to the 6500. DUMP3 then sets and enables a 3 minute

time-out in RTCLK, enables AMC transmit interrupts, and returns via DAMC to the interrupted program.

When the line is correctly received by the 6500 inputting program, the checksums match, and then the program sends a ":" to the Dump Mode. DUMP3 is called, detects the ":", resets the time-out, and then formats the next line of 81 ASCII characters. This line contains the remainder of the sensor data points, represented again as pairs of ASCII characters. If all the data points in the sensor data block are added to the output buffer before it contains 78 characters, DUMP3 checks the sensor ID words to locate the next defined sensor at the same station. If there are none, it begins checking the sensor ID words at the next defined station. When it finds the next defined sensor, it calls DDISK to read the sensor data block and the sensor definition block from the disk. DUMP3 then adds two ASCII "a"s to the output buffer, followed by the new station number and sensor number, the twelve multiplier and offset ASCII characters, and the ASCII-coded sampling interval, as before. The ASCII representations of the data points are then added to the output buffer until it contains 78 characters. DUMP3 calls CKSUM again, sets up the input and output block instructions and the 3 minute time-out in RTCLK, and then enables the time-out and the AMC interrupts. DUMP3 returns and the interrupted program resumes executing. This process continues until all the defined sensor data blocks have been transmitted to the 6500.

When the two ASCII characters representing the last data point from the last defined sensor at the last defined station have been added to the output buffer, DUMP3 fills the remainder of the output buffer (thru character 78) with "a"s. DUMP3 then sets the end flag,

DENDFL, and calls CKSUM. The input and output block instructions and the 3 minute time-out are set up, and then the time-out and AMC transmit interrupts are enabled. DUMP3 returns, and the interrupted program resumes executing. When the next AMC receive end-of-block interrupt occurs, and the character received is a ":", DUMP3 writes the termination character, '(:EO), and a carriage return into the output buffer. DUMP3 then writes the address of DUMP4 into location DRCV, sets up the input and output block instructions and the 3 minute time-out, and then returns thru DRCV to the interrupted program. The inputting program executing on the 6500 terminates after it receives the :EO code. The 6500 then sends the "READY" message. As each character is received, DMAC calls DUMP4, which calls DREADY. As explained above, DREADY checks each character as it is received and returns to a different location in DUMP4 when all the characters of the "READY" message, except the last one, have been received. DUMP4 then writes the address of a second entry point, DUMP4B, into location DRCV and returns to DAMC, which restores machine status and returns to the interrupted program. When the last character of the "READY" message is received, DAMC calls DUMP4B, which resets the time-out in RTCLK and resets the disk section flag to indicate that the section was successfully transmitted. DUMP4B then sets up a block output instruction to send a command to the 6500. This command instructs the 6500 to execute a program which adds the sensor data just transferred to the 6500 data base. DUMP4B then writes the address of DUMP5 at location DRCV and sets up a block input instruction to generate an end-of-block interrupt when one character is received. DUMP4B then enables a 30 minute time-out and the AMC transmit interrupts and returns to DRCV, which restores

machine status and returns to the interrupted program.

After the 6500 data base program finishes executing, the 6500 sends the "READY" message to the Dump Mode. DUMP5 is called by DAMC as each character is received, and in turn calls DREADY. DREADY again returns to a different location in DUMP5 when all the characters of the "READY" message, except the last one, have been received. DUMP5 then sets the address of a second entry point, DUMP5B, into location DRCV and returns to DAMC, which restores machine status and returns to the interrupted program. When the final character of the "READY" message is received, the AMC receive end-of-block interrupt calls DAMC, which calls DUMP5B. DUMP5B resets the 30 minute time-out in RTCLK and sets up a block output instruction to send the logout message to the 6500. DUMP5B also writes the address of DUMP6 into location DRCV. DUMP5B then sets up a block input instruction which generates an input end-of-block interrupt when 30 characters have been received from the 6500. DUMP5B sets and enables a 2 minute time-out in RTCLK, enables AMC transmit interrupts, and returns thru DAMC to the interrupted program. After the logout message has been transmitted, the AMC transmit end-of-block interrupt calls DAMC1, which enables AMC receive interrupts. When the 30 characters of the logout message have been received, the AMC receive end-of-block interrupt calls DAMC, which then calls DUMP6. DUMP6 resets the time-out in RTCLK and resets modem/calling unit #2. DUMP6 then calls DMPMGR, which checks if any more sections of the disks are filled. If they are, DMPMGR sets up and enables a call request to the ACU to dial up the 6500. DMPMGR then returns to DUMP6, which returns thru DAMC to the interrupted program. If all the filled sections were transferred, DMPMGR resets the modem/calling unit #2 busy flag,

M2BSY, and returns to DUMP6, which again returns thru DAMC to the interrupted program. The Dump Mode terminates until DMPMGR is again called by DASMGR.

An exception interrupt, signifying either a broken telephone connection or incorrect data transmission, may occur at any time. This interrupt calls DAMC2. DAMC2 first saves machine status and then inputs the status word from the AMC. If the telephone connection was broken, and the exception enable flag, DEXCEN, is set, DAMC2 resets modem/calling unit #2 and then sets the time-out in RTCLK for one second. DAMC2 restores machine status and returns to the interrupted program. When one second elapses, RTCLK calls DTO, which calls DMPMGR to set up another telephone call to the 6500. The entire section of 24 hours of all the sensor data is then retransmitted. If a data transmission error occurs or if the exception enable flag, DEXCEN, is reset, DAMC2 simply restores the machine status and returns to the interrupted program. The data transmission error is subsequently detected by the time-out in RTCLK or by the checksum procedure.

Each time that DMPMGR is called it checks if DMPFLG is still set. RTCLK resets this flag variable at 20:00. If DMPMGR is called and DMPFLG is reset, it releases modem/calling unit #2 and terminates the Dump Mode. The Dump Mode will resume when DMPMGR is called by DASMGR at 9:00 the next weekday morning. A flow chart of the Dump Mode is shown in Figure 4.3.

4.9 Disk Error Mode

This Mode is entered if an error occurs in one of the disk units. The disk controller detects the error, and then the routine

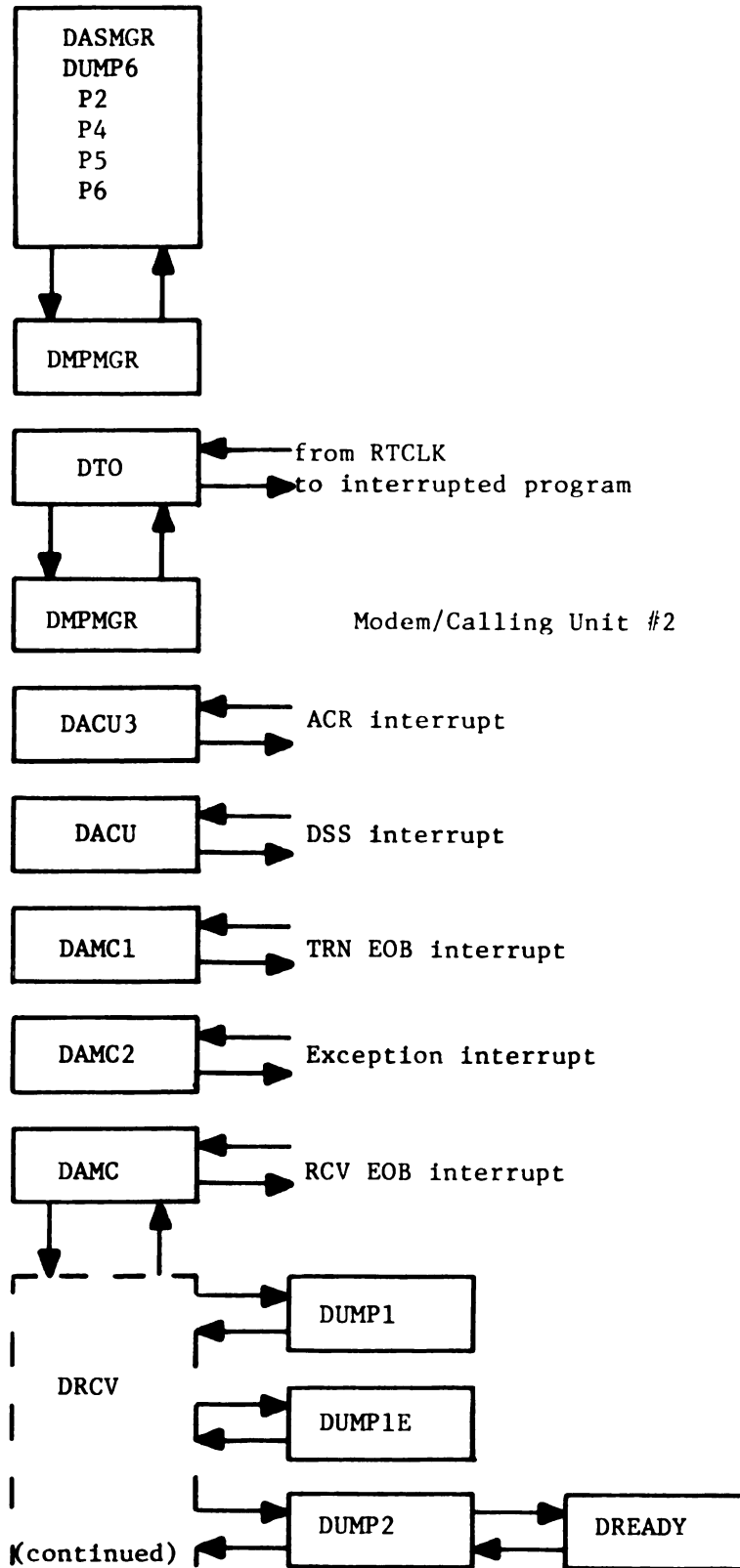


Figure 4.3 Dump Mode Flow Chart

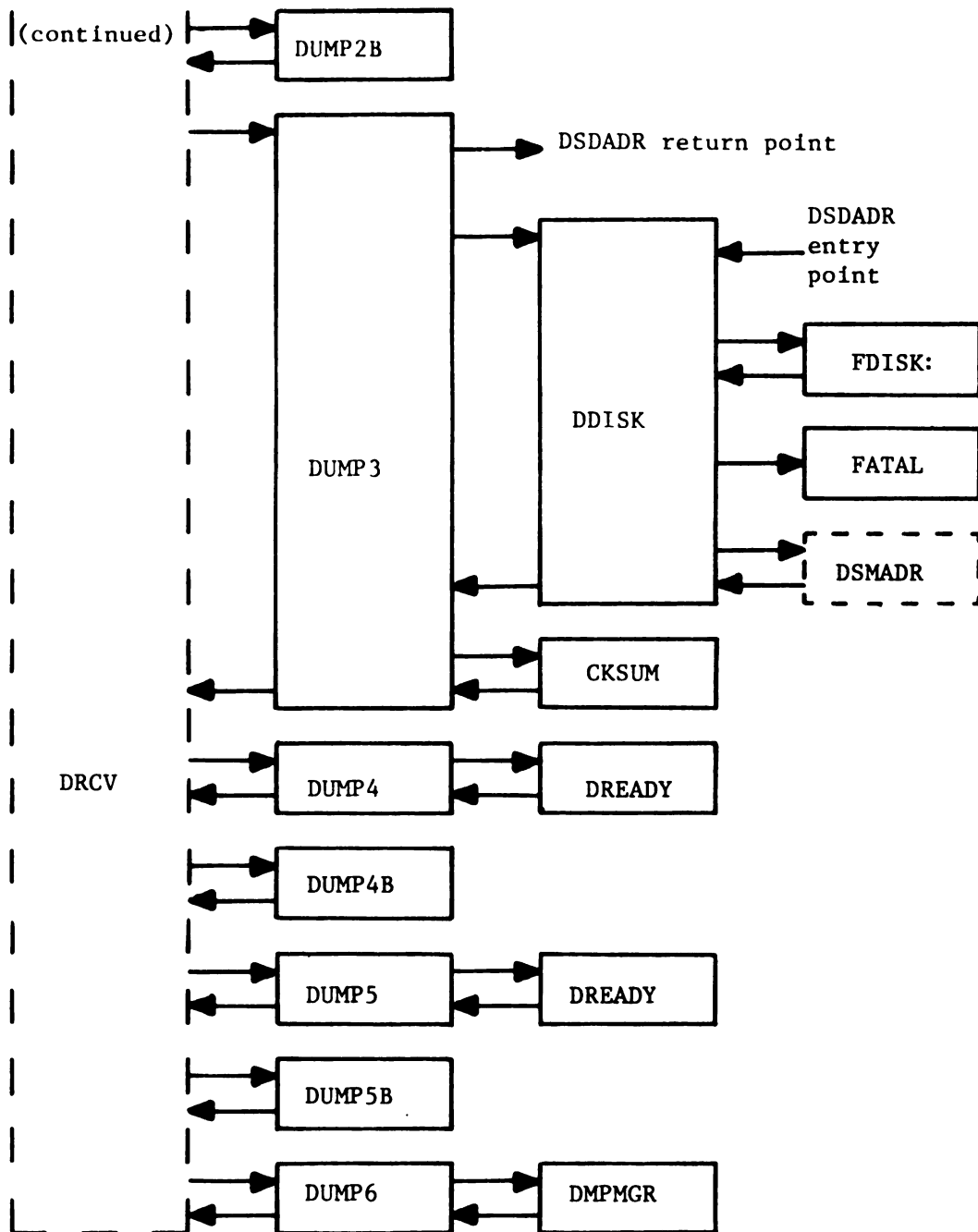


Figure 4.3 (cont'd)

accessing the disk calls FATAL. FATAL reads the disk status word and outputs it to the minicomputer console data register so that the type of error may be determined. See Appendix B for the error codes. FATAL then turns on the OV lite, which is also on the minicomputer console, if disk unit 1 caused the error, and turns the OV lite off if disk unit 0 caused the error. Both modem/calling units are then reset, and the TTY controller is initialized. All other Modes are terminated until the disk error condition is corrected and the DAS has restarted itself.

FATAL next loads the A register with :C and executes an OTA :15,3 instruction. This turns on the audible alarm at the Central Station and lights the red FATAL ERROR indicator lights on the Local Teletype and on the Central Station Display Panel. The alarm may be silenced by momentarily pushing the flashing red ALARM STOP push button on the Display Panel. The FATAL ERROR indicator lights remain on. FATAL now turns on the LINE BUSY indicator light on the Local Teletype (SEL :13,3), enables output to the Remote Teletype (SEL :13,6), and turns on the Local Teletype (SEL :15,0). FATAL then calls PFR1 to print each line of the Fatal System Crash message on both Teletypes. The first line contains the control codes :93 and :90, which turn on the Remote Teletype and light the LINE BUSY indicator light on the Remote Teletype, respectively. PFR1 writes the address of PFR2 into location MPOUT, sets up a block output instruction, and resets PRNTFL. PFR1 then enables TTY output interrupts and continuously checks PRNTFL. When the block of characters is finished printing, the TTY controller generates an end-of-block interrupt which calls TTYOUT. TTYOUT saves machine status and calls PFR2, which sets PRNTFL. PFR2 returns to TTYOUT,

which restores machine status and then returns. PFR1 detects that PRNTFL is set and returns to FATAL. Each line of the message is printed by a call to PFR1. The final line sent to the Teletypes contains the control codes :94 and :97, which turn off the Remote Teletype and the Remote LINE BUSY indicator light, respectively. FATAL then turns off the Local Teletype (SEL :15,1) and resets the Local LINE BUSY indicator light (SEL :13,4). FATAL next instructs the modem/controller to send a constant space to the modem/controller at the Remote Teletype (SEL :13,2). The remote modem/controller senses this constant space and turns off the green RUN indicator light on the Remote Teletype and lights the red FATAL ERROR indicator light.

FATAL then continuously checks the status of the blue RESET push button switch on the Display Panel of the Central Station. After the cause of the disk error is determined and corrected, the operator pushes the RESET button. FATAL detects this and calls the disk controller to check each disk drive. If either disk unit now generates an error, FATAL again sends the unit number and the error code to the minicomputer console and reactivates the alarm. The disk controller rechecks both disk units when the RESET button is again pushed. Once both disk drives check out successfully, FATAL turns off the FATAL ERROR indicator lights on the Local Teletype and the Display Panel and turns on the yellow START indicator lights by setting the A register to 2 and executing an OTA :15,3 instruction. FATAL then turns off the TTY carrier (SEL :13,0), which causes the remote modem/controller to light the yellow START indicator light on the Remote Teletype. FATAL next calls PWRUP, which restarts the DAS as if a power failure had occurred. A flow chart of the Disk Error

Mode is shown in Figure 4.4.

4.10 Power Restart Mode

The Power Restart Mode is entered when power is restored following a power failure. All other Modes are suspended until the DAS has restarted itself. The minicomputer processor detects when a power failure is about to occur and generates a Power-Down interrupt. This interrupt calls PWRDN, which disables the recognition of any more interrupts. When power is restored to an acceptable level, the processor generates a Power-Up interrupt, which calls PWRUP.

PWRUP first checks if the Disk Error Mode was active when the power failed. If it was, FTLFLG is set, and PWRUP then calls FATAL thru the FTLENT entry point. Otherwise PWRUP turns the remote TTY carrier off (SEL :13,0), which causes the remote modem to light the yellow START indicator light on the Remote Teletype. PWRUP then turns on the START indicators on the Local Teletype and the Display Panel of the Central Station by setting the A register to 2 and executing an OTA :15,3 instruction. PWRUP next checks PFFLG to determine if the Power Restart Mode itself was active when the power failure occurred. If it wasn't, PWRUP saves the time and date that the power failure occurred. Otherwise PWRUP uses the time computed when PWRUP was executing before the most recent power failure. PWRUP then resets the time variables in RTCLK to zero and sets PFSKIP. The real-time clock is then enabled; PWRUP continuously checks MIN until it is incremented, which occurs in exactly one minute. PWRUP waits one minute before doing any restart processing in case another power failure should occur immediately after the original one. After the minute has elapsed, PWRUP resets the real-time clock and turns on the

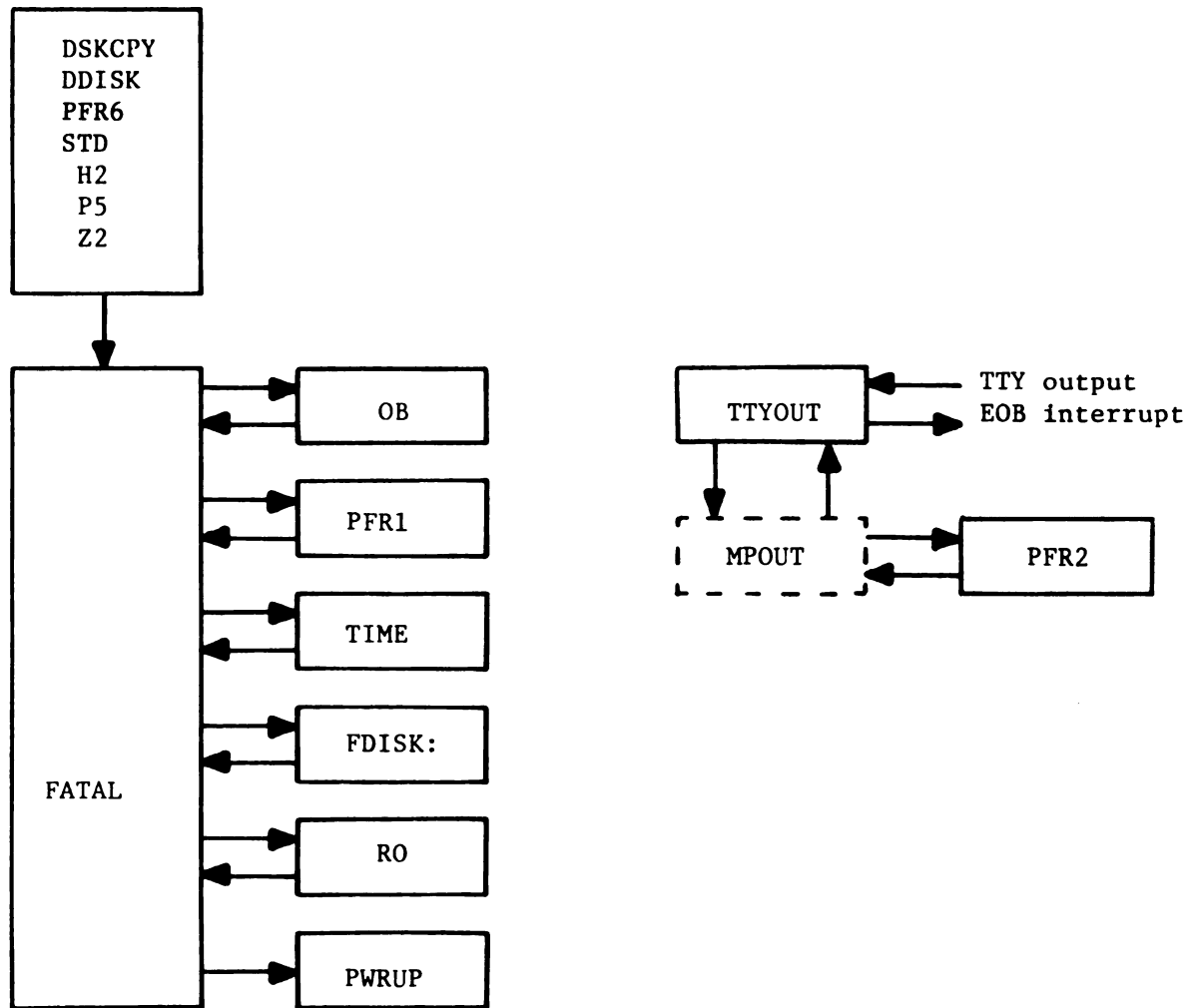


Figure 4.4 Disk Error Mode Flow Chart

remote TTY carrier (SEL :13,1).

PWRUP next prints the Power Failure message on both Teletypes. The first line of the message includes the control code to turn the Remote Teletype on (:91). The Local Teletype is automatically turned on when system power is restored. PWRUP calls PFR1 to print each line of the message. When the entire message is printed, PWRUP resets all three disk flags (DSDREQ, DSMREQ, and DSKBSY) and then checks DCPFLG to determine if the Automatic Acquisition Mode was copying section 1 of the disk when the power failure occurred. If a copy was in progress, PWRUP calls DSKCPY to redo the copy. PWRUP then resets DCPFLG and DASFLG.

If the Power Restart Mode was active before the power failed, PWRUP now checks if it was updating the sensor data blocks on section 1 of the disk when the power failure occurred. If it was, PWRUP calls PFR6. PFR6 adds the correct number of missing-data flags to all defined sensor data blocks to update them to the Restart Time computed before, which is stored in RSMIN, RSHOUR, and RSDAY. PFR6 uses the same routines used by the Automatic Acquisition Mode to access the disk and update the sensor data blocks. When PFR6 complete the update, it returns to PWRUP. If PWRUP was copying section 1 of the disk to another section when the power failure occurred, it now repeats the copy operation by calling DSKCPY.

When these operations are completed, or if PWRUP wasn't active before the power failure, it reads the current time from the battery-operated back-up clock by calling BTIME. PWRUP then computes how much time elapsed while the power was off and updates the time variables used by RTCLK to keep the DAS time. PWRUP then calls BATUPD to set the newly computer time into the battery-operated

clock. This is necessary since the day-of-the-year counter in the battery-operated clock must be reset if a new year started during the power failure.

PWRUP then checks if the power was off long enough for the Automatic Acquisition Mode to miss a sampling time. If no sampling times were missed, PWRUP starts up the real-time clock and jumps to finish the restart procedure. However, the Automatic Acquisition Mode may have been active when the power failure occurred or the Automatic Acquisition Mode may have missed a sampling time. In these cases PWRUP determines how much time was lost and then begins one of three possible restart sequences.

Restart Sequence 1 is executed if the power was off at 9:00, when section 1 of the disk should have been copied into another section by the Automatic Acquisition Mode. PWRUP first calls PFR3, which sets the Restart Time (RSMIN, RSHOUR, and RSDAY) to 9:15 and then enables the real-time clock. PFR3 returns and then PFR6 is called. PFR6 adds a sufficient number of missing-data flags to each sensor data block on section 1 of the disk so that each block is updated to 9:00. PFR6 then returns, and PWRUP calls PFR7. PFR7 copies the entire section 1 of the disk into an unused section so that the Dump Mode may later transfer the sensor data to the 6500 Computer system. PFR7 calls DSKCPY to do the actual copy operation. PFR7 returns, and then PWRUP calculates the actual Restart Time. This Restart Time is ahead of the current DAS time. When the actual time reaches the Restart Time, RTCLK enables the Automatic Acquisition Mode every quarter hour. After the Restart Time is computed, PWRUP calls PFR6, which again adds the correct number of missing-data flags to all the sensor data blocks on section 1 of the disk to

update them to the Restart Time. PFR6 returns and PWRUP jumps to finish the restart procedure.

Restart Sequence 2 is executed if the 9:00 section 1 copy operation was not missed while the power was off. PWRUP first starts up the DAS real-time clock and then calls PFR4 to set the actual Restart Time. PFR4 returns, and then PFR6 is called to update section 1 of the disk. As before, a sufficient number of missing-data flags are added to each sensor data block to account for the number of samples that the Automatic Acquisition Mode missed during the power outage. When PFR6 returns, PWRUP jumps to finish the restart procedure.

Restart Sequence 3 is executed if more than one 9:00 section 1 disk copy operation was missed. In this case power was off for more than 24 hours. PWRUP first calls PFR3 to set the Restart Time to 9:15 and enable the real-time clock. PFR3 returns, and then PWRUP calls PFR6 to add enough missing-data flags to the sensor data blocks on section 1 of the disk, as before. PFR6 returns, and then PFR7 is called. PFR7 copies section 1 of the disk to another section for the Dump Mode and then returns. PWRUP next calls PFR4 to set the actual Restart Time. PFR4 returns, and then PWRUP calls PFR6, which fills each sensor data block with missing-data flags. PFR6 returns, and then PWRUP finishes the restart procedure.

PWRUP finishes up the restart procedure by resetting all the disk and modem flag variables and initializing the Automatic Acquisition Mode error buffer. PFSKIP is then reset and PFSCAN set, so that RTCLK checks the Restart Time against the actual DAS time. When the two times match, RTCLK resets PFSCAN and then sets DASFLG every quarter hour. PWRUP next prints a message on both Teletypes

indicating that the Power Restart is complete. PFR1 is called to print each line of the message and returns to PWRUP as each line finishes printing. After the last line is printed, PWRUP calls OFF to turn both Teletypes off. Operator access via the Manual Mode is now possible. PWRUP then sets DUMPEN if the DAS time is later than 9:00 and earlier than 19:00. This enables DASMGR to call DMPMGR to begin the Dump Mode, if it is a weekday. PWRUP then enters the Automatic Acquisition Mode sense loop. It continuously checks the flag variable DASFLG and calls DASMGR to supervise the Automatic Acquisition Mode when DASFLG is set. RTCLK sets DASFLG every quarter hour, once the actual DAS time reaches the Restart Time. A flow chart of the Power Restart Mode is shown in Figure 4.5.

4.11 Manual Mode

A Manual Mode access to the DAS may be initiated from either the Local Teletype or the Remote Teletype. Both Teletypes are connected to the TTY port of the minicomputer processor option card. Logic circuits on the Custom Card allow the software to select which Teletype will be active. These software commands are detailed in Chapter 2. A Manual Mode access begins when an operator pushes the REQUEST button on either one of the Teletypes. These REQUEST buttons are designed to send a control character to the processor. The Local REQUEST button sends :11, while the Remote REQUEST button sends :12. The processor receives this input and generates a TTY input end-of-block interrupt. This interrupt calls TTYIN, which first saves the status of the machine and then resets the TTY input interrupts so they cannot occur until set again. TTYIN next executes a jump and store instruction to the address contained in location MPIN. This

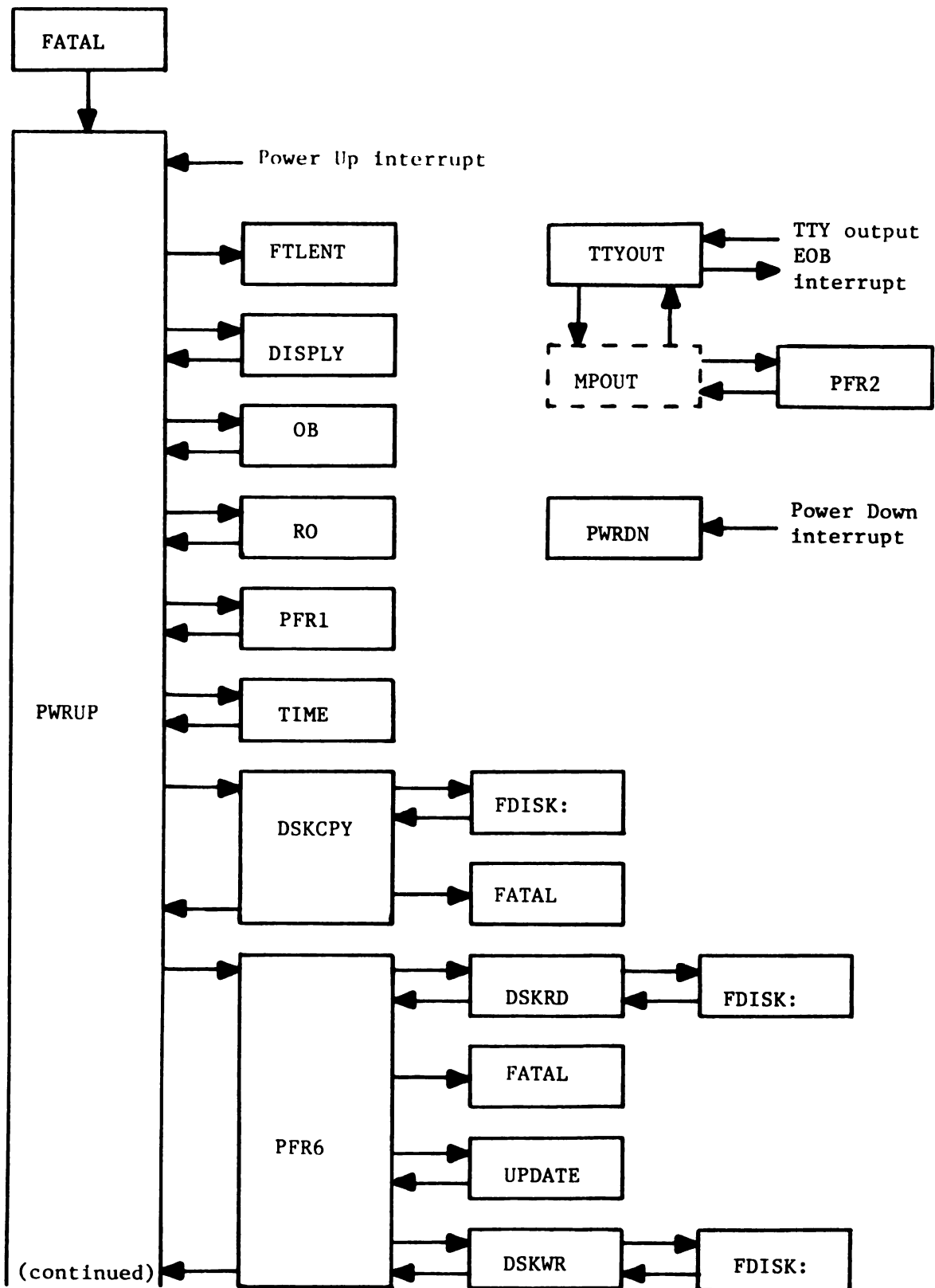


Figure 4.5 Power Restart Mode Flow Chart

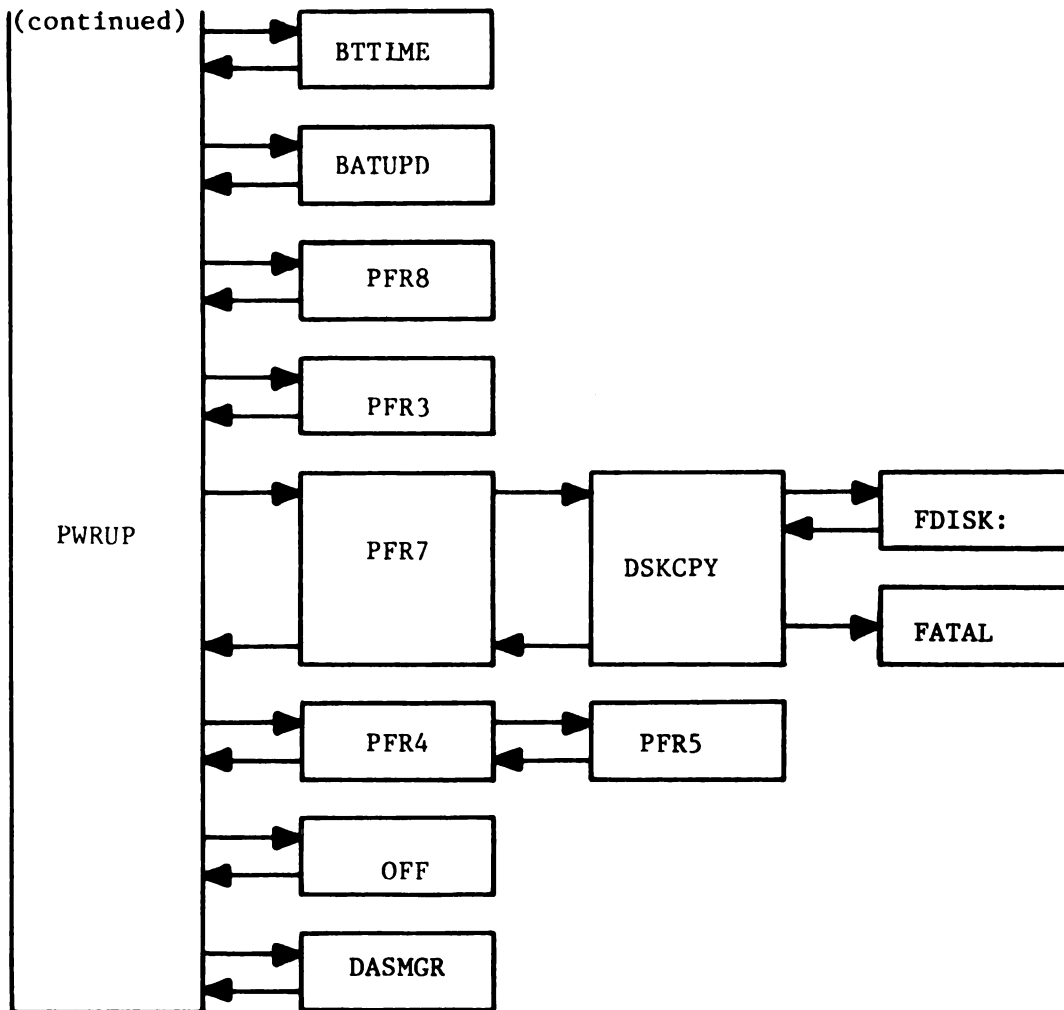


Figure 4.5 (cont'd)

location has previously been set with the starting address of routine ON. ON therefore begins executing and checks the character just inputted thru the TTY port. TTYFLG is set to 1 if the Local REQUEST button was pushed and set to 2 if the Remote REQUEST button was pushed. ON also resets both Level 1 and Level 2 authorization keys.

If the Local Teletype requested access, ON turns the Local Teletype on (SEL :15,0) and enables output to the Remote Teletype (SEL :13,6). It then sets up a block output instruction to send six control characters (:90) to the Remote Teletype. This control character turns on the LINE BUSY indicator light on the Remote Teletype. Six characters are sent to allow the Local Teletype motor to come up to full speed before the prompt asterisk is printed. The blue REQUEST button on the Local Teletype lights up to indicate that the Teletype is on. ON also sets the address of a second entry point, ON4, at location MPOUT. ON then enables the TTY output interrupts and returns to TTYIN. TTYIN restores machine status and returns to the interrupted program. After all six characters are sent to the Remote Teletype, a TTY output end-of-block interrupt is generated by the TTY controller, and TTYOUT is called. It saves the machine status, disables the TTY output interrupts so they cannot occur until set again, and then executes a jump and store instruction to the address contained in location MPOUT. MPOUT is defined in routine TTYOUT and was set with the address of ON4. ON is entered thru the ON4 entry point and enables the Local Teletype (SEL :13,5). It also stores its return address at the first location of PROMPT and then jumps to the second location of PROMPT.

If the Remote Teletype requested access, ON turns the local LINE BUSY indicator light on (SEL :13,3) and enables output to the Remote

Teletype (SEL :13,6). It then sets up a block output instruction to send six characters (:93). This is the control code to turn on the Remote Teletype. Six control characters are sent to allow the Remote Teletype motor to come up to full speed before the prompt asterisk is printed. As with the Local Teletype, the REQUEST button lights up when the Remote Teletype is on. ON then sets the address of PROMPT in location MPOUT, enables TTY output interrupts, and returns to TTYIN. TTYIN restores machine status and returns to the program in progress when the interrupt occurred. After all the characters are sent to the Remote Teletype, a TTY output end-of-block interrupt is generated, and TTYOUT is called. Machine status is saved, and then PROMPT is called since its address was written in location MPOUT.

PROMPT begins executing and loads the output buffer with a carriage return, line feed, and a number of rubouts. The number of rubouts which follow a carriage return is dependent on the type of terminal used. For the standard Teletype, two rubouts are sufficient to allow the carriage to return before beginning the next line of type. The variable ROFLG is set to one-half the number of rubouts desired. Thus ROFLG equals one for Teletypes. Following the rubouts, PROMPT then adds a control character to enable auto-echo at the Teletype (:95), an asterisk and a space. The Teletype controller on the minicomputer processor option card always operates in the full-duplex mode, with its auto-echo feature suppressed. Sending :95 to either Teletype enables an auto-echo feature at that Teletype, so that whatever is typed there is also automatically printed. PROMPT then sets up a block output instruction and stores the address of a second entry point, PR4, at location MPOUT. The TTY output interrupts are enabled, and PROMPT returns to TTYOUT. TTYOUT

restores machine status and returns to the interrupted program.

When all the characters are sent to the appropriate Teletype, a TTY output end-of-block interrupt is generated, and TTYOUT is called. It saves machine status and calls PROMPT thru the PR4 entry point. PROMPT then sets up a block input instruction which gives an input end-of-block interrupt after a single character is typed on the Teletype. PROMPT also sets the address of SCAN at location MPIN and enables a 30 second time-out in RTCLK. If 30 seconds elapse before anything is typed on the Teletype, RTCLK calls OFF, which then turns off the Teletype and terminates the Manual Mode access to the DAS. PROMPT then enables TTY input interrupts and returns to TTYOUT, which restores machine status and returns to the interrupted program.

When a character is typed on the Teletype, TTYIN is called and saves machine status; SCAN then begins executing. It inputs the character just typed and checks if it is a valid command letter. If an invalid command letter was typed, SCAN stores its own return address at the first location of PROMPT and then jumps to the second location of PROMPT, which requests another command, as described above.

If a period or escape (ESC) was entered, SCAN calls OFF. OFF resets TTYFLG and checks if the Manual Mode was using either modem/calling unit and resets it if necessary. It then enables output to the Remote Teletype (SEL :13,6) and sets up a block output instruction to turn the Remote Teletype off (:94) and to reset the LINE BUSY indicator light on the Remote Teletype (:97). OFF also sets the address of a second entry point, OF4, at location MPOUT. OFF enables TTY output interrupts, returns to SCAN, which returns to TTYIN, which then returns to the interrupted program. After the two characters are

sent to the Remote Teletype, a TTY output end-of-block interrupt is generated. TTYOUT is called and saves machine status, and then enters OFF thru the OF4 entry point. It resets the local LINE BUSY indicator light (SEL :13,4), turns the Local Teletype off (SEL :15,1), and sets the address of routine ON at location MPIN. OFF then enables TTY input interrupts so that routine ON will be called when one of the REQUEST buttons is pushed. OFF returns to TTYOUT, which returns to the interrupted program.

If one of the commands that doesn't require a password is requested, SCAN calls that routine immediately. For commands that require a Level 1 password, SCAN saves the beginning address of the command routine in local variable MANENT, and then checks if either the Level 1 or Level 2 password has already been entered during this Manual Mode access. Commands requiring Level 2 authorization also have their starting address saved in MANENT, but SCAN checks only if the Level 2 password was already entered. If the correct password was entered, SCAN calls the command routine by executing a jump and store instruction to the address contained in location MANENT. However, if the correct password was not already entered, SCAN calls the K routine to request it. Entry to K is made thru the K1V entry point for the Level 1 password and thru the K2V entry point for the Level 2 password.

K sets its starting address in MPIN and MPOUT so that it will be called when a TTY input or output end-of-block interrupt occurs. K then calls KEY, which sets up the block output instruction to print the message on the Teletype requesting the operator to enter the password. KEY enables the TTY output interrupts and returns to K, which returns to SCAN. SCAN then returns to TTYIN, which restores

machine status and returns to the interrupted program.

When the password request message is finished printing, an output end-of-block interrupt is generated, and K is called from TTYOUT. K then calls KEY, which sets up a block input instruction to accept the password, enables the 30 second time-out in RTCLK and the TTY input interrupts, and then returns to K. K returns to TTYOUT, which returns to the interrupted program. When four characters have been typed by the operator, the TTY input end-of-block interrupt is generated, and K is called from TTYIN. K then calls KEY, which checks the password and returns to K. If the password was valid, K loads the starting address of the command routine, and calls the command routine. For an invalid password, K sets up a block output instruction to print the "INVALID PW" message on the Teletype and sets the address of PROMPT in location MPOUT. K then enables the TTY output interrupts and returns to TTYIN, which returns to the interrupted program. After the "INVALID PW" message is finished printing, PROMPT requests a new command, as explained before. A flow chart of the Manual Mode is shown in Figure 4.6.

4.11.1 A Command

The A command sets sensors and stations on-line at the Central Station. They are then automatically sampled by the Automatic Acquisition Mode. The A command begins executing by entering routine AD thru the A entry point. ADDFLG is then set to indicate that the requested sensors will be placed on-line.

When AD is first called, the starting address of AD is stored in locations MPIN and MPOUT so that the routine will be called when either a TTY input or output end-of-block interrupt occurs. AD then

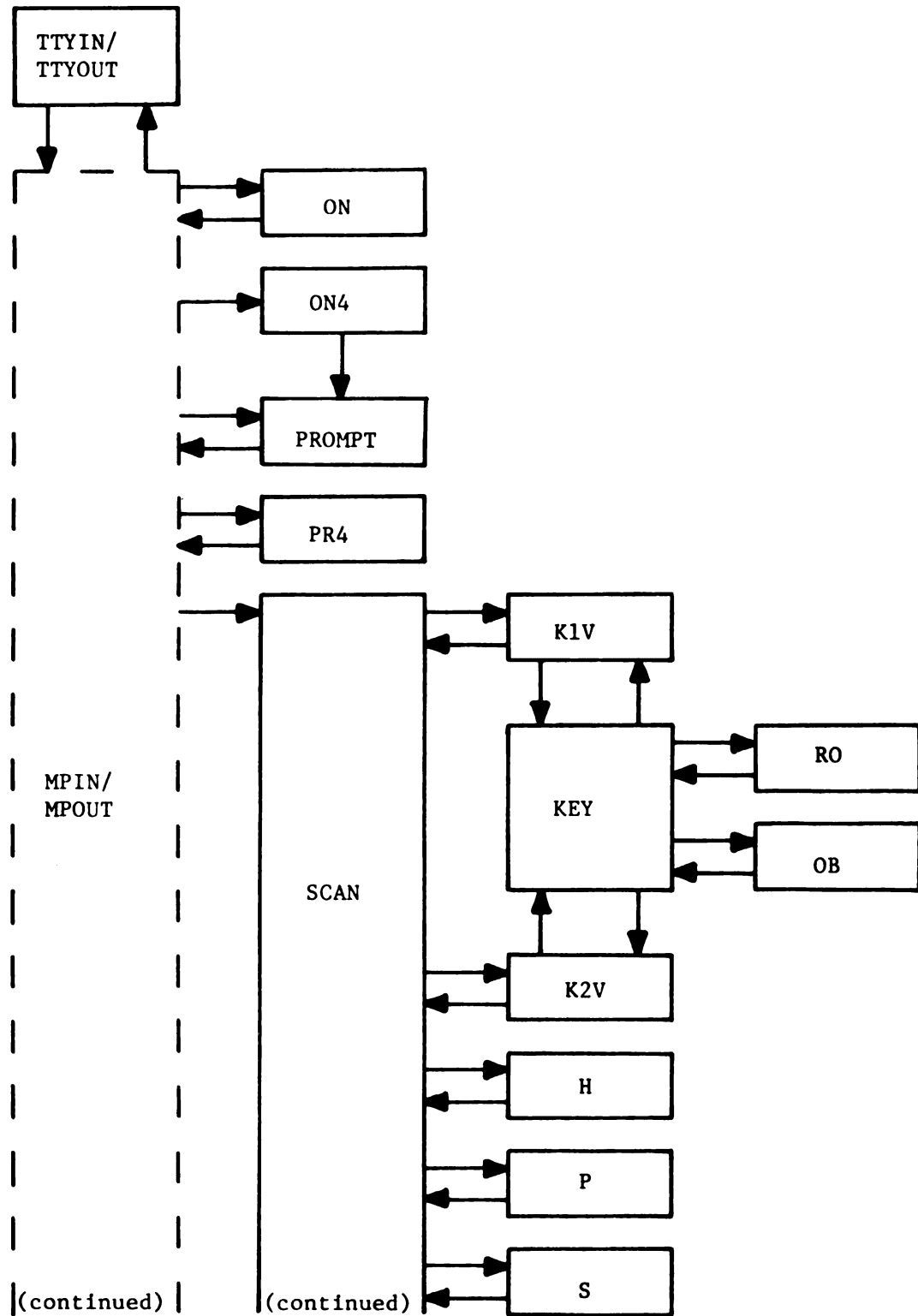


Figure 4.6 Manual Mode Flow Chart

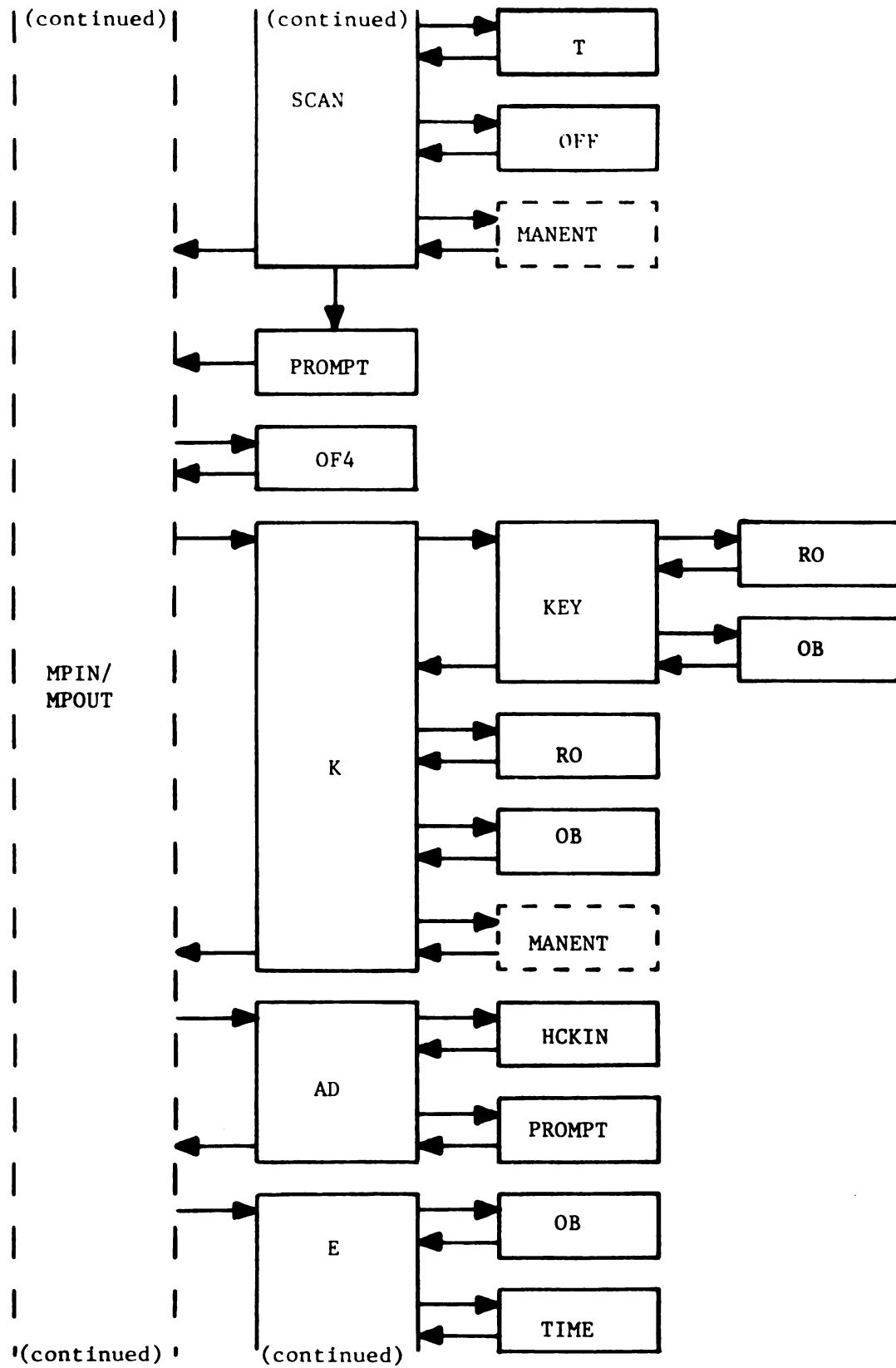


Figure 4.6 (cont'd)

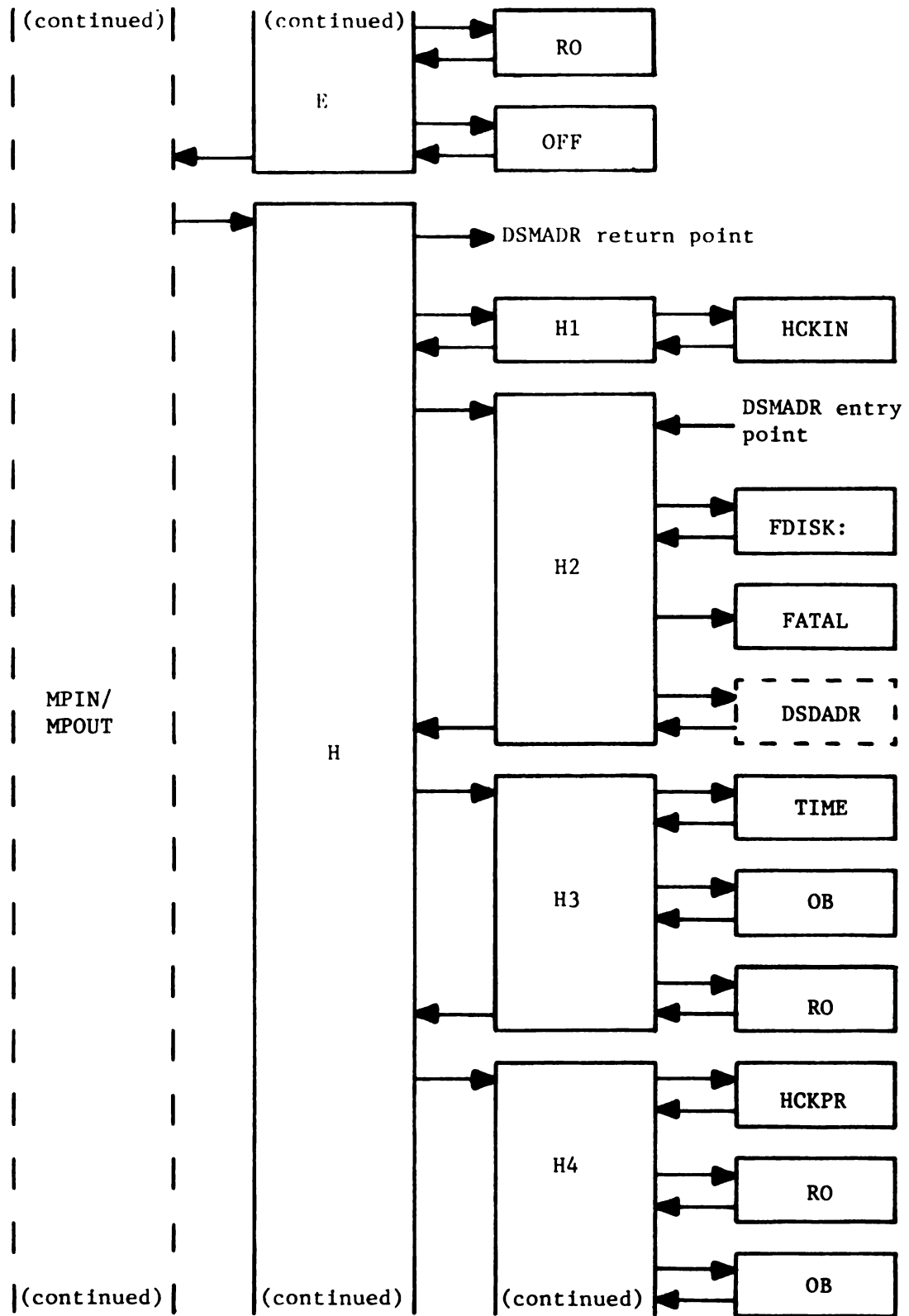


Figure 4.6 (cont'd)

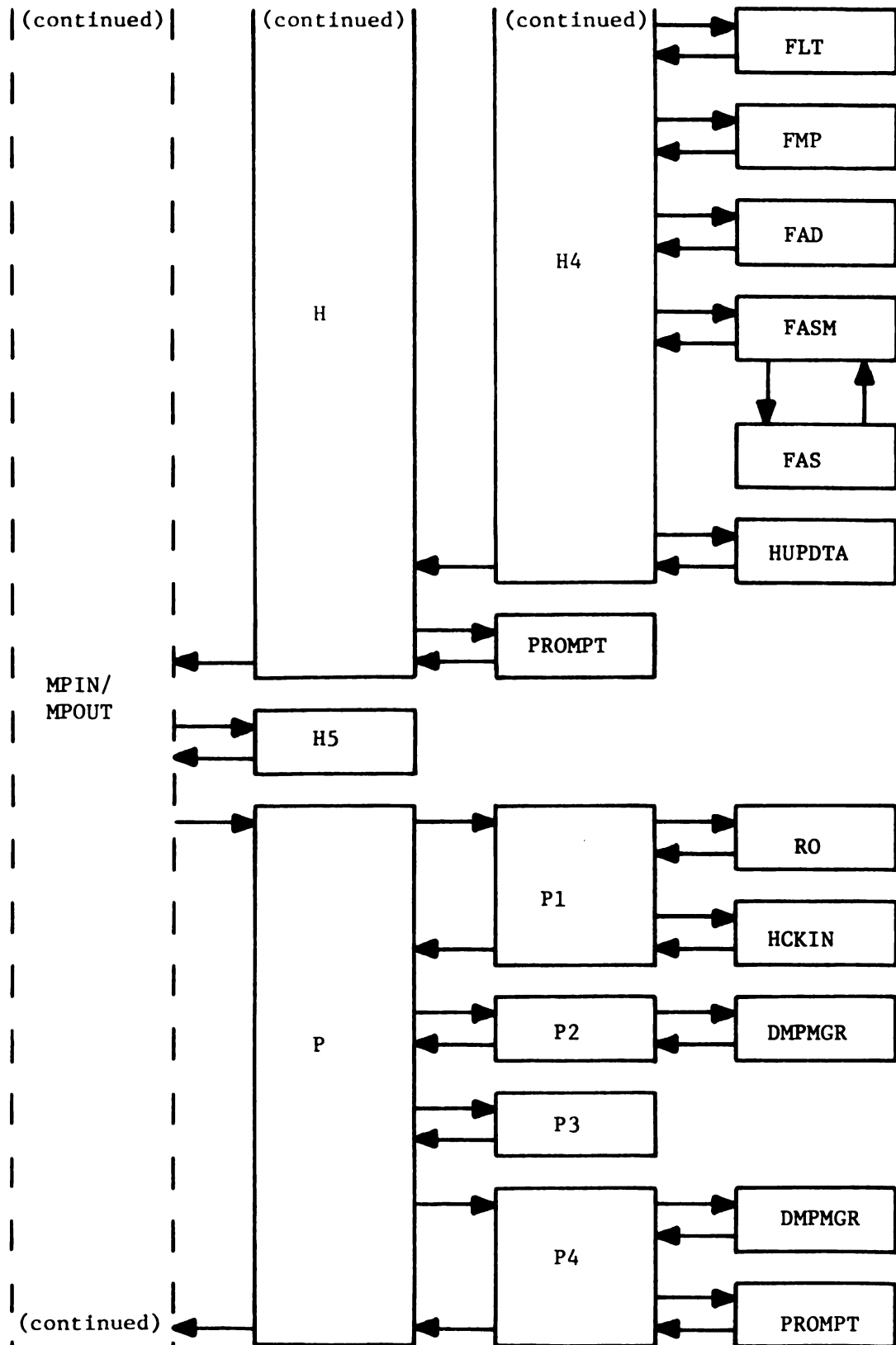


Figure 4.6 (cont'd)

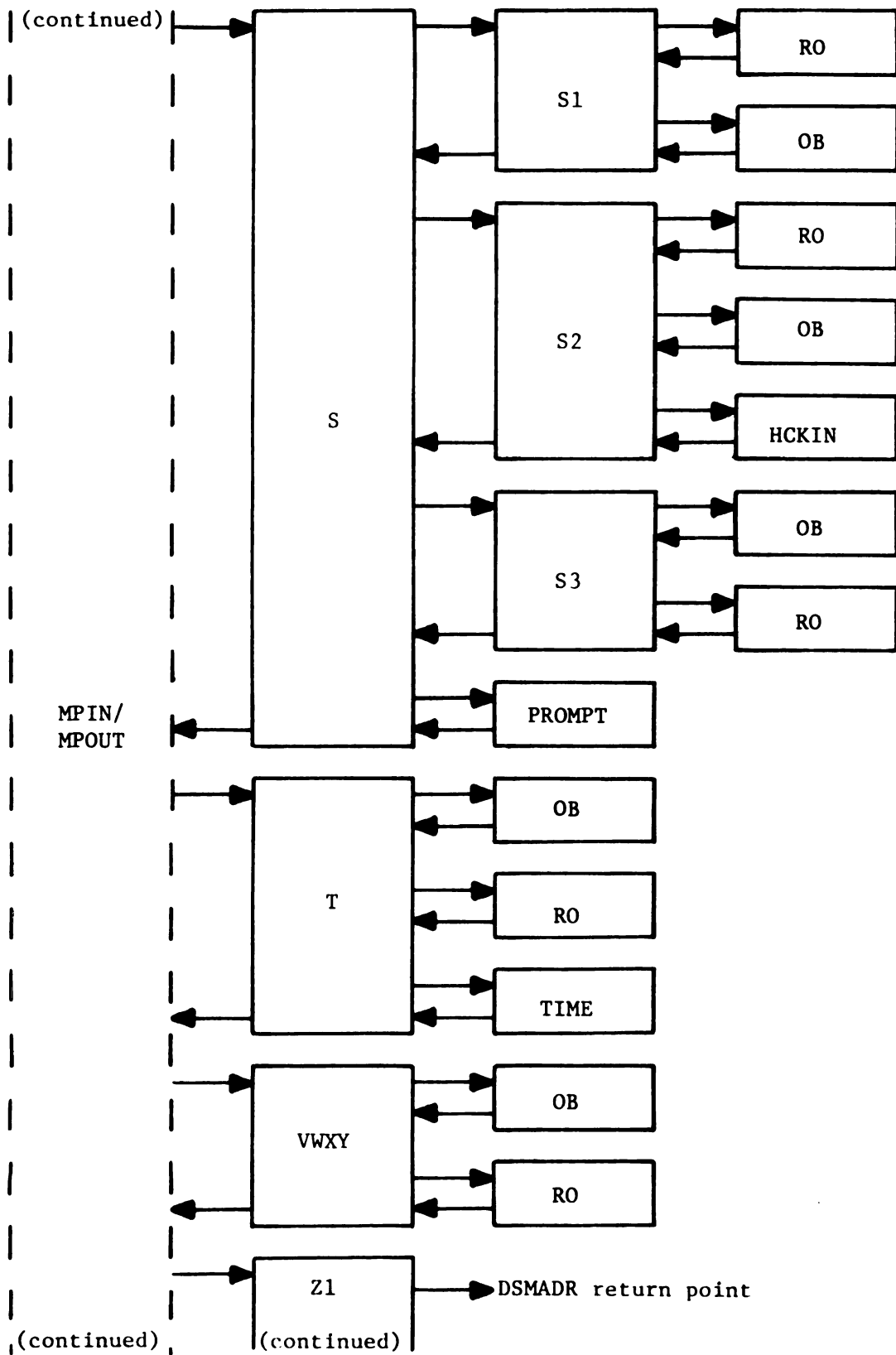


Figure 4.6 (cont'd)

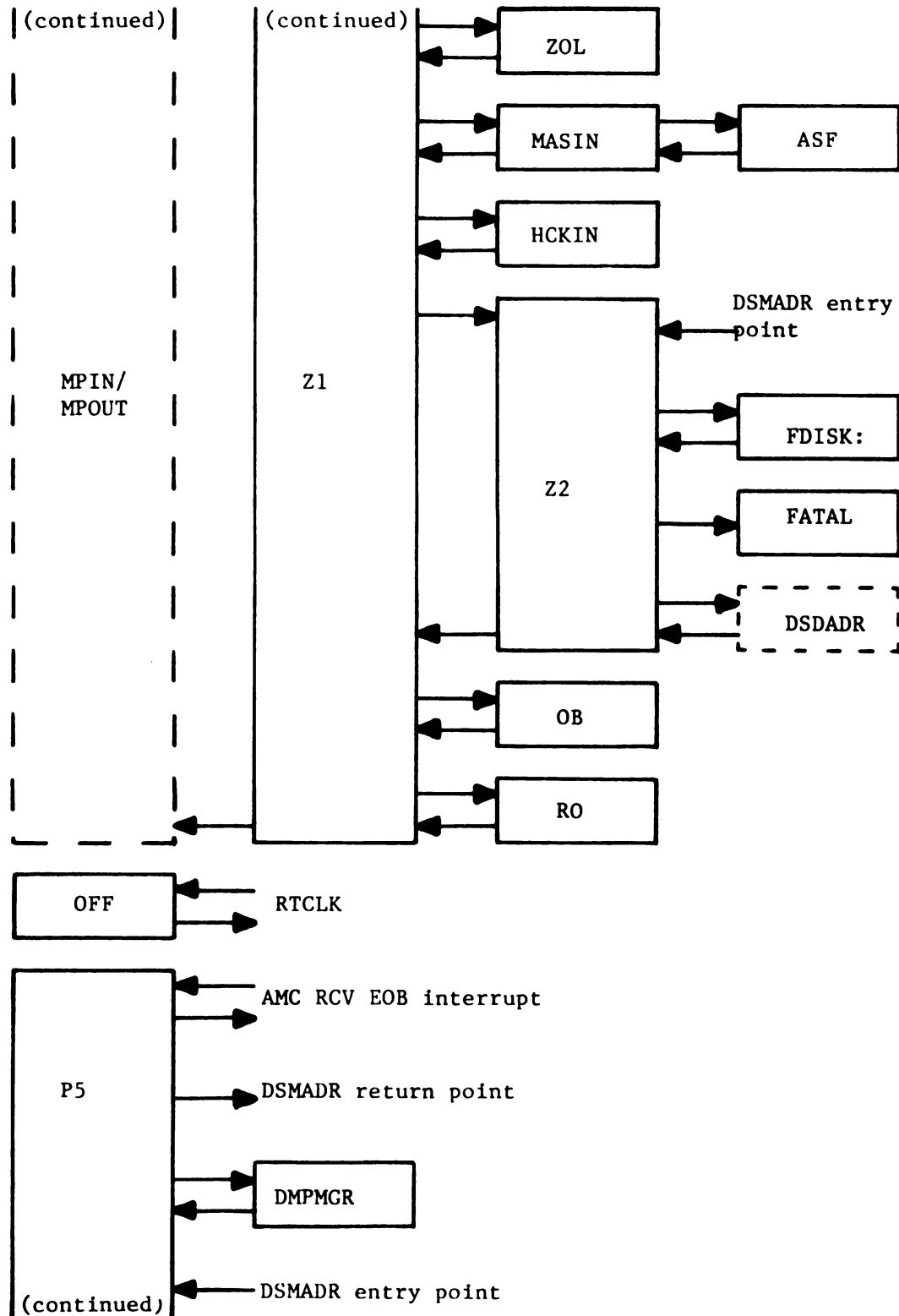


Figure 4.6 (cont'd)

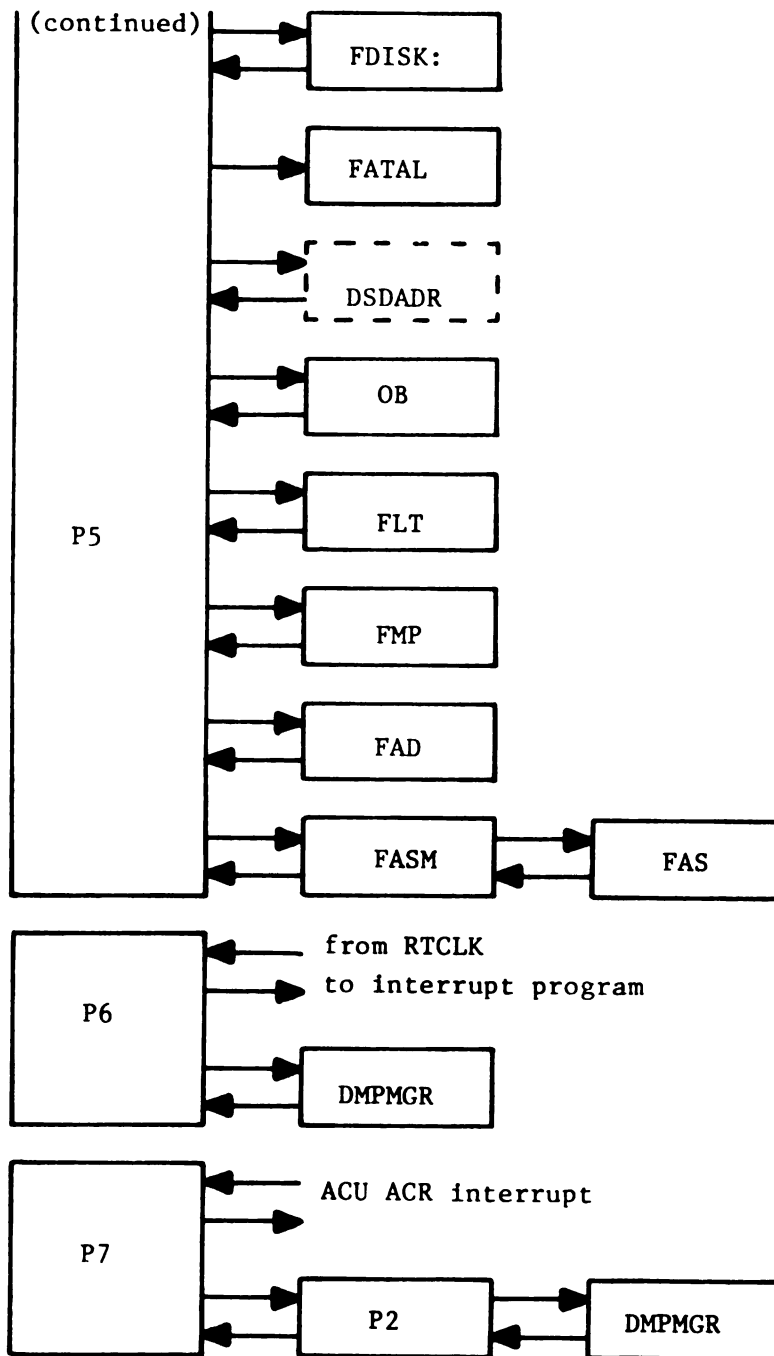


Figure 4.6 (cont'd)

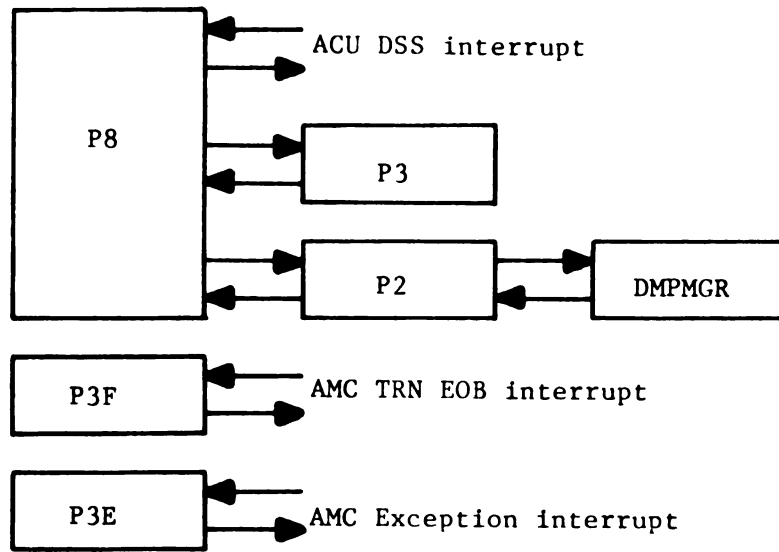


Figure 4.6 (cont'd)

sets up a block output instruction to print the "STA= " message. It also sets up a block input instruction to generate an input interrupt when one character is typed on the Teletype. AD then writes the code for the instruction to disable the TTY output interrupts (SEL 7,7 = :403F) at the location of the TTY output end-of-block interrupt. Thus, when the block of output finishes printing, the end-of-block interrupt only resets the output interrupt, and TTYOUT is not called. The interrupted program continues immediately after the reset instruction is executed. AD next enables the 30 second time-out in RTCLK and the TTY input and output interrupts and returns. The interrupted program then resumes executing.

After the message is finished printing and the operator types a single character, the TTY input end-of-block interrupt calls TTYIN, which calls AD. AD then calls HCKIN, which checks the character just typed. If an escape (ESC) or a period was typed, HCKIN indicates this by returning to a different location in AD. In this case, AD calls prompt, which requests a new command, as before. AD then returns thru TTYIN to the interrupted program. If a digit was typed, HCKIN stores it and returns to AD, which sets up a block input instruction to input another single character. AD then returns to TTYIN, which returns to the interrupted program. HCKIN deletes the most recently entered character when a back-arrow (←) is typed. This process continues until three characters, not including deleted characters, are entered or until a space is typed, whichever comes first. HCKIN then forms the digits into a binary number and returns to a different location in AD. HCKIN recognizes if any non-numeric characters were entered and returns to a still different location; AD then sets up a block output instruction to repeat the request for a

station number, and the whole process repeats itself. AD also repeats the request for a station number if the station requested isn't defined.

Once a correct station number is completely entered, AD sets up another block output instruction to request the sensor number. It also sets up a block input instruction, as before, to accept a single character from the Teletype. AD then enables TTY input and output interrupts and returns to TTYIN, which returns to the interrupted program. The input interrupt again causes TTYIN to call AD, which calls HCKIN. The same steps that were used to input the station number are then repeated for the sensor number, with one exception. If the letter A is typed, HCKIN indicates this to AD, and then AD adds all the sensors at the requested Remote Station.

After the sensor number is completely entered and verified, AD places the requested sensor(s) on-line by resetting bit 0 of the appropriate sensor ID words in core memory. AD then sets up a block output instruction to request the next station number and returns to TTYIN, which returns to the interrupted program. This process continues until the operator types an escape (ESC) for a station or sensor number or a period for a station number.

4.11.2 D Command

The D command sets sensors and stations off-line at the Central Station. When it is time for the Automatic Acquisition Mode to sample those sensors which are off-line, a missing-data flag is written into their sensor data blocks.

The D command begins executing by entering AD thru the D entry point. ADDFLG is reset to indicate that the requested sensors will

be placed off-line. AD then executes as explained for the A command. The requested sensors are placed off-line by setting bit 0 of the sensor ID words in core memory.

4.11.3 H Command

The H command prints the sensor data which was collected by the Automatic Acquisition Mode during the previous 24 hours for an operator-selected group of sensors. The data from a maximum of six sensors may be listed in a single printout.

When the H command is first called, it calls H1, which initializes various flags and also sets the address of H in locations MPOUT and MPIN. H1 then sets up a block output instruction to print the message "STA= " and a block input instruction to generate an interrupt after one character is typed on the Teletype. H1 also writes the code for the instruction to disable the TTY output interrupts (SEL 7,7 = :403F) at the location of the TTY output end-of-block interrupt, as was done for routine AD. H1 enables TTY input and output interrupts and then returns to H. The interrupted program resumes executing. After the message is printed and a single character entered, the TTY input end-of-block interrupt calls TTYIN, which calls H, which in turn calls H1. H1 then calls HCKIN, which inputs the character and checks it. The characters for the station and sensor numbers are inputted as described for the A command. After the complete sensor number is verified, H1 checks how many station/sensor number pairs have been entered. If six pairs were entered, H1 writes the address of H5 into location MPIN, sets up a block input instruction, and then enables the TTY input interrupts. H5 is called after a single character is typed; it sets HESFLG if the

character is an escape and then returns to the interrupted program. HESFLG is checked in H4 to see if the listing should be terminated. After enabling the TTY input interrupts, H1 returns to a different location in H, which then calls H2. This same sequence of instructions is also followed if a period, rather than a space, terminates the previous sensor number. Otherwise H1 requests more station and sensor numbers.

H2 checks the disk flag, DSKBSY, to see if the disk controller is being used by another Mode. If the disk controller is busy, H2 sets the Manual Mode request flag, DSMREQ, and writes the address of a second entry point into H2 at location DSMADR. H2 then returns to H, which returns to the interrupted program. When the other Mode finishes using the disk controller, it checks DSMREQ to see if a request is in. Since there is one, the other Mode executes a jump and store instruction to the address at location DSMADR. This transfers control to H2, which writes the return address to the other Mode into the first word of routine H. H2 then reads the sensor definition blocks and sensor data blocks from disk unit 0 for each sensor for which a history listing is requested. H2 also copies the sensor data block for each of the requested sensors to a storage area in core memory. When this is completed, H2 checks the Dump Mode disk request flag, DSDREQ. If the Dump Mode has requested the disk, H2 executes a jump and store instruction to the address at location DSDADR. Control is then transferred to the Dump Mode. When the Dump Mode is finished with the disk, it returns to H2, which then returns to H. H calls H3, which sets up a block output instruction to print the time and date the history was requested. H3 enables TTY output interrupts and returns to H. H returns, and the interrupted program

then resumes executing. When the line finishes printing, the TTY output end-of-block interrupt calls TTYOUT, which calls H, which calls H3. H3, via the same steps, prints two more lines: one with the labels for each of the requested sensors and the second with the sensors' units. H3 also sets up several flags which are used by the H4 routine.

After the line containing the sensors' units finishes printing, the TTY output end-of-block interrupt calls TTYOUT, which calls H. H checks if HESFLG has been set by an escape (ESC) typed on the Teletype. If so, it calls PROMPT to request a new command and then returns thru TTYOUT to the interrupted program. If HESFLG isn't set, H calls H4. H4 calls HCKPR, which checks the time of the most recent data from the sensors that are to be listed. H4 then assembles an output block containing this time, as well as the data for those sensors that were sampled at the time. The raw sensor data is obtained from the individual sensor data blocks, which were copied into core memory by H2. The multiplier and offset are obtained from the sensor definition blocks, which were also copied into core memory by H2. H4 calls FLT to convert the raw sensor data into a floating point number and then calls FMP to multiply the result by the multiplier. FAD then adds the offset to this result. FASM is called to convert the final floating point number into ASCII characters which can be printed. Each sensor data point is processed in this manner. When a full line is computed, H4 calls HUPDTA to update the flags that H4 uses to keep track of which data point from each sensor is up next for printing. H4 then sets up a block output instruction, enables TTY output interrupts, and returns to H. H then returns to TTYOUT, which returns to the interrupted program.

When this line of output finishes printing, the TTY output end-of-block interrupt calls H thru TTYOUT. H checks HESFLG and calls H4 to format the next line of output if termination wasn't requested by the operator. If HESFLG is set, H calls PROMPT to request a new command and then returns thru TTYOUT to the interrupted program. When the history listing exceeds a full page, H4 adds enough line feeds after the last printed line to create a new page. H4 then sets a flag in the H routine so that the sensor labels and units will be printed by H2 when H is called the next time. When H4 sets up the output block for the final line of a history listing, it returns to a different location in the H routine. H then loads the address of PROMPT into location MPOUT, so that a new command will be prompted for when the output end-of-block interrupt occurs.

4.11.4 K Command

The K command allows the operator to enter a new Level 1 password. It was previously described how the Level 1 password was requested from the operator and verified by the K routine. This description assumes that authorization has already been obtained to execute the K command. The K routine is then entered thru the KIM entry point. K first stores its starting address at location MPIN and MPOUT. It then calls KEY, which sets up a block output instruction to print the message requesting the new password. KEY enables TTY output interrupts and returns to K. K returns, and the interrupted program resumes executing. After the message is finished printing, the TTY output end-of-block interrupt calls TTYOUT, which calls K, which then calls KEY again. KEY now sets up a block input instruction which generates an input end-of-block interrupt

after four characters are typed on the Teletype. KEY enables the 30 second time-out in RTCLK and the TTY input interrupts and returns to K. K returns to TTYOUT, which returns to the interrupted program.

When the four characters are typed, the TTY input end-of-block interrupt calls TTYIN, which then calls K. K now checks the four characters entered to insure that they are only alphabetic or numeric characters. If all four characters satisfy this condition, they become the new Level 1 password. K then sets up a block output instruction to print the message "OK". If illegal characters were entered, K sets up a block output instruction to print the message "INVALID PW". K then writes the address of PROMPT into location MPOUT and enables TTY output interrupts. K returns to TTYIN, which then returns to the interrupted program. The PROMPT routine requests a new command, as explained before, when the message finishes printing.

4.11.5 P Command

The P command prints the current value of any specified sensor. A telephone connection to the correct Remote Station is automatically established, and the requested sensor is then sampled. The correctly scaled sensor value, including its label and units, is printed on the Teletype.

The P command begins executing by calling P, which calls P1. P1 sets the address of P in locations MPIN and MPOUT and initializes flags used by the P routines. P1 then requests the operator to enter a station and sensor number via the same sequence of instructions the AD routine uses. Once a valid station and sensor number are entered, P1 saves them and returns to P. P then calls P2, which checks if a

modem/calling unit is already being used by the Manual Mode. If it is not, P2 checks the modem/calling unit busy flags, M1BSY and M2BSY, to determine if they are being used by the Dump Mode or the Automatic Acquisition Mode. If both modem/calling units are in use, P2 writes the address of PROMPT in location MPOUT and sets up a block output instruction to print the message "BOTH PHONE LINES IN USE" on the Teletype. P2 then enables TTY output interrupts and returns to P, which returns to TTYIN, which returns to the interrupted program. After the message finishes printing, PROMPT requests a new command.

One of the modem/calling units may be available; in this case, P2 sets that modem/calling unit's busy flag and then writes the addresses of the Manual Mode Automatic Calling Unit (ACU) and Asynchronous Modem Controller (AMC) interrupt routines into the correct interrupt locations in the base page. The base page interrupt locations are listed in Appendix C. If a modem/calling unit is already attached to the Manual Mode, P2 checks if it is connected to the requested station. If it is, P2 returns to P, which then calls P3. If not, or if the modem/calling unit was just attached to the Manual Mode, P2 reads the station definition block in core memory to get the telephone number of the requested Remote Station. P2 then initializes the ACU and the AMC and checks that the telephone line is clear. If it isn't, P2 does a 3 second time-out and checks it again. If the line is still occupied (telephone handset off-hook), P2 sets up a block output instruction to print the message "LINE 1 OCCUPIED" or "LINE 2 OCCUPIED", depending on which line was occupied. P2 then enables TTY output interrupts and returns to P, which returns to TTYIN, which returns to the interrupted program. After the message finishes printing, the TTY output end-of-block interrupt calls

113

TTYOUT, which calls P. P then calls P4, which releases the modem/calling unit that was being used by the Manual Mode. If modem/calling unit #2 was attached to the Manual Mode, P4 checks M2DREQ to see if the Dump Mode has a request in for it. If it does, P4 executes a jump and store instruction to DMPMGR. After DMPMGR finishes executing, it returns control to P4, which returns thru TTYOUT to the interrupted program.

After the unoccupied telephone line is initialized, P2 sets up a block output instruction to send the digits comprising the Remote Station's telephone number to the ACU. It then enables the ACU and returns to P, which returns thru TTYIN to the interrupted program. The call to the Remote Station may be unsuccessful, due to a busy number or no answer within approximately 40 seconds. These conditions cause the ACU to generate an Abandon Call and Retry (ACR) interrupt. P7 is called immediately and saves the machine status. It then resets the correct modem/calling unit and calls P2. P2 sets up a block output instruction to print the message "STATION UNAVAILABLE", enables TTY output interrupts, and returns to P7. P7 then restores machine status and returns to the interrupted program. When this message finishes printing, P1 is called and requests another station and sensor number, as explained above.

A Data Set Status (DSS) interrupt may also be generated by the ACU. This interrupt calls P8 directly. P8 saves the machine status and then checks that all the digits in the telephone number were dialed. If they were not, P8 resets the modem/calling unit and then calls P2, which again sets up a block output instruction to print the message "STATION UNAVAILABLE". P2 then enables TTY output interrupts and returns to P8, which restores machine status and returns to

the interrupted program. P1 again requests another station and sensor number when the message finishes printing, as above.

Provided all the digits were dialed, P8 calls P3. P3 first sets up a block output instruction which sends a string of characters to the Remote Station to request it to sample the desired sensor. P3 then sets up a block input instruction to receive the five characters sent back from the Remote Station and enables a 3 second time-out in RTCLK. P3 enables AMC receive and transmit interrupts and returns to P8. P8 restores machine status and returns to the interrupted program.

In case three seconds elapse before the sensor data is received from the Remote Station, RTCLK calls P6. P6 transfers the machine status saved in RTCLK and then resets the AMC receive interrupt. P6 next checks if either the Dump Mode or the Automatic Acquisition Mode have requested the attached modem/calling unit. If the Automatic Acquisition Mode has a request in, P6 simply resets the modem/calling unit and the request flag, M2AREQ. However, if the Dump Mode requests the attached modem/calling unit, P6 resets it and then executes a jump and store instruction to DMPMGR. After DMPMGR finishes executing, it returns to P6, which sets up a block output instruction to print the message "NO RESPONSE FROM SENSOR". P6 then enables TTY output interrupts, restores machine status, and returns to the interrupted program. After the message finishes printing, P1 is called to request another station and sensor number.

Usually the five characters containing the sensor data are received from the Remote Station within three seconds. The AMC receive end-of-block interrupt then calls P5 directly. P5 saves machine status and checks if either the Dump Mode or the Automatic

Acquisition Mode have requested the attached modem/calling unit. If so, the request is handled as explained in the above paragraph. Otherwise, P5 sets the correct modem/calling unit idle flag, M1IDLE or M2IDLE. This allows the Dump Mode or the Automatic Acquisition Mode to seize the modem/calling unit from the Manual Mode without requesting it first. P5 then checks if the sensor just sampled is the same one previously sampled by the P command. If it isn't, P5 checks the status of the disk controller. In case the controller is busy, P5 follows the same procedure outlined for the H command to gain access to the disk. When P5 does have access, it reads the correct sensor definition block from disk 0 to obtain the label, units, multiplier, and offset for the sensor that was just sampled. P5 then formats the output line, which contains the sensor's label, the correctly scaled value of the sensor, and the units. P5 next sets up a block output instruction to print the line, enables TTY output interrupts, restores machine status, and returns to the interrupted program. After the line has finished printing, the TTY output end-of-block interrupt calls TTYOUT, which calls P. P then calls P1 to request another station and sensor number.

The Manual Mode is terminated if the P command has to wait more than 30 seconds for the operator to type a character. RTCLK detects when 30 seconds have elapsed and calls OFF. OFF executes as explained before. The operator may intentionally terminate the P command by typing a period when a station number is requested or by typing an escape (ESC) in place of either the station or sensor number. P1 then returns to P, which calls P4 to terminate the P command. P4 checks the other Modes to see if they have requested the attached modem/calling unit and handles their requests as explained

above. P4 next calls PROMPT to request a new command and then returns to P. The interrupted program resumes executing.

4.11.6 S Command

The S command lists all the defined Remote Stations and the defined sensors at particular operator-selected Remote Stations. The S command begins executing by calling S, which calls S1. S1 writes the address of S into location MPOUT and then checks each of the station definition blocks in core memory to determine which Remote Stations are defined. For each defined station, S1 checks the sensor ID words in core memory to determine if all the defined sensors at that station are off-line at either the Central Station or the Remote station itself. S1 then sets up a block output instruction to list those stations which are defined. If all the sensors at a station were placed off-line at the Remote Station, the station number is followed by a single asterisk. If all the sensors were placed off-line at the Central Station (by the D command), S1 types two asterisks after the station number. A maximum of ten station numbers are listed on one line. S1 then enables TTY output interrupts and returns to S. The interrupted program resumes executing.

When the output end-of-block interrupt occurs, TTYOUT calls S, which calls S1 again if a second line of station numbers must be printed. S1 sets up this output block, enables TTY output interrupts, and returns to the interrupted program. Otherwise, if all the defined station numbers are printed, S calls S2, which sets up the block instructions to request a station number from the operator, as described for the A command. After a valid station number is entered, S2 returns to S, which calls S3. S3 sets the address of

routine H5 at location MPIN and enables TTY input interrupts. If an escape is typed by the operator, HESFLG is set in H5. S3 next sets up a block output instruction to list each of the sensors defined at the requested station. Again, a single asterisk follows a sensor number if that sensor was placed off-line at the Remote Station (bit 1 of the sensor ID word in core memory set) and two asterisks if it was placed off-line at the Central Station (bit 0 of the sensor ID word set). Ten sensor numbers are listed per line. S3 then enables TTY output interrupts and returns to S, which returns thru TTYOUT to the interrupted program.

The TTY end-of-block interrupt calls TTYOUT when the line finishes printing, and TTYOUT then calls S. S checks HESFLG to see if the operator typed an escape while the line was being printed. If HESFLG is set, S calls PROMPT to request a new command and then returns thru TTYOUT to the interrupted program. If HESFLG isn't set and more sensors at the requested station have to be listed, S calls S3, which sets up the next line of output, as before. When the last line of sensor numbers is finished printing, the TTY output end-of-block interrupt calls TTYOUT, which then calls S. S now calls S2 to request another station number.

The S command can also be terminated by typing an escape when a station number is requested. This causes S2 to return to S, which then calls PROMPT. PROMPT requests a new command and returns to S, which returns thru TTYOUT to the interrupted program.

4.11.7 T Command

The T command prints the DAS time and date on the Teletype. The T command begins executing by calling T. T calls TIME, which reads

the time variables (MIN, HOUR, DAY, MTH, and YEAR). TIME then fills the output buffer with the ASCII representations of these variables and returns to T. T sets up a block output instruction and writes the address of PROMPT in location MPOUT. T then enables TTY output interrupts and returns. The interrupted program resumes executing; PROMPT requests a new command when the line finishes printing.

4.11.8 U Command

The U command allows the operator to enter a new Level 2 password. The procedure the DAS uses to request the Level 2 password was previously explained; it is assumed that Level 2 authorization has already been obtained to execute this command. The U command begins executing by entering the K routine thru the K2M entry point. The U command then executes the same sequence of instructions that the K command does to change the Level 1 password.

4.11.9 V Command

The V command defines a new Remote Station. It begins executing by entering the VWXY routine thru the V entry point. VWXY sets its beginning address at location MPOUT and then sets up a block output instruction to send a carriage return, line feed, and rubouts to the Teletype. VWXY enables TTY output interrupts and returns. The interrupted program then resumes executing. When the TTY output end-of-block interrupt occurs, TTYOUT calls VWXY, which then sets up another block output instruction to print the message "DEFINE STATION". VWXY enables TTY output interrupts and returns to the interrupted program thru TTYOUT. This message finishes printing, and the TTY output end-of-block output interrupt calls TTYOUT, which again calls VWXY. This time VWXY sets the address of routine Z1 at

locations MPOUT and MPIN and then sets up a block output instruction to send a carriage return, line feed, and rubouts to the Teletype. VWXY also sets a flag variable, ZRFLG, that is used by Z1 to determine which section it should execute. VWXY then enables TTY output interrupts and returns to the interrupted program thru TTYOUT.

Z1 is called from TTYOUT when the TTY output end-of-block interrupt occurs and sets up a block output instruction to print the message "STA= ". Z1 also sets up a block input instruction to input a single character from the Teletype. Z1 then sets and enables a 3 minute time-out in RTCLK, enables TTY input and output interrupts, and returns thru TTYOUT to the interrupted program. The station number is read in, character by character, as described for the A command, with the exception that the time-out in RTCLK is always set for 3 minutes. The station number is then checked; if it's already defined, Z1 sets up a block output instruction to print the "ABORT" message and writes the address of PROMPT at location MPOUT. Z1 then enables TTY output interrupts and returns. The interrupted program resumes executing; the PROMPT routine requests a new command when the "ABORT" message finishes printing.

In the event the Remote Station isn't already defined, Z1 sets up a block output instruction to send a carriage return, line feed, and rubouts to the Teletype. Z1 enables TTY output interrupts and returns thru TTYIN to the interrupted program. When the TTY output end-of-block interrupt occurs, Z1 is again called from TTYOUT and sets up another block output instruction to print the message "PHONE= ". Again, Z1 sets up a block input instruction to read one character from the Teletype. TTY input and output interrupts are

enabled, and Z1 returns thru TTYOUT to the interrupted program. Z1 is then called as each character is typed. When Z1 is called after a space is typed, it sets up a block output instruction to send a carriage return, line feed, and rubouts to the Teletype. Z1 then enables TTY output interrupts and returns thru TTYIN to the interrupted program. When the TTY output end-of-block interrupt calls Z1 thru TTYOUT, it sets up another block output instruction to print the message "OK?" and also sets up a block input instruction to read a single character from the Teletype. Z1 enables TTY input and output interrupts and returns thru TTYOUT to the interrupted program.

After a single character is typed, Z1 is called from TTYIN. If a "Y" wasn't typed, Z1 sets up a block output instruction to print the "ABORT" message and writes the address of PROMPT into location MPOUT. TTY output interrupts are then enabled, and Z1 returns to TTYIN, which returns to the interrupted program. Typing a "Y" causes Z1 to add the new Remote Station to the DAS. First Z1 computes the number of digits that were entered for the Remote Station's telephone number and writes this number, along with the telephone number itself, into the correct station definition block in core memory. Z1 then sets up a block output instruction to print "ADDED", writes the address of PROMPT at location MPOUT, and enables TTY output interrupts. Z1 returns, and the interrupted program resumes executing. When the message finishes printing, PROMPT begins executing and requests a new command.

4.11.10 W Command

The W command removes a Remote Station. It begins executing by entering the VWXY routine thru the W entry point. VWXY then executes

the same sequence of instructions that it does for the V command. The two exceptions are that the message printed for the W command is "REMOVE STATION" and ZRFLG is set for a different entry point into Z1.

After the message finishes printing, Z1 begins executing as described for the V command. However, Z1 only requests the station number for the W command. After the station number is completely entered, Z1 verifies that the requested Remote Station is already defined. In the instance it is not, Z1 aborts and calls PROMPT to request a new command, as described for the V command. Otherwise Z1 asks if it should go ahead and remove the Remote Station. Again, the sequence of instructions is the same as those executed for the V command. Typing "N" aborts Z1 and calls PROMPT, while typing "Y" causes Z1 to check if either the Dump Mode or the Automatic Acquisition Mode is processing data from the requested Remote Station. If this is the case, Z1 aborts the W command. The W command may then be attempted again in a few minutes. If neither Mode is processing data from the requested Remote Station, Z1 removes it by writing :FFFF into its disk address word in core memory. Z1 then sets up a block output instruction to print the message "DELETED" and writes the address of PROMPT at location MPOUT. Z1 enables TTY output interrupts and returns thru TTYIN to the interrupted program. When the message is finished printing, PROMPT begins executing and requests a new command.

4.11.11 X Command

The X command defines a new sensor. It begins executing by entering the VWXY routine thru the X entry point. VWXY then executes

as described for the V command, with the exception that the message printed on the Teletype is "DEFINE SENSOR". ZRFLG is also set for a different entry point into Z1.

Z1 then begins executing and prints each line and inputs the typed characters as described for the V command. After the station number and sensor number are completely entered, Z1 checks the sensor disk address word in core memory to determine if the sensor is already defined. If this is the case, Z1 aborts, and the DAS then prompts for a new command, as explained for the V command. Otherwise, if the sensor doesn't already exist, Z1 requests the sampling interval ("3-SI= ") and the disk address of the sensor data block ("4-CYL= ", "5-SEC= ", "6-WORD= "). The limits on these parameters are given in the description of the X command in Chapter 3. After the complete disk address is read in, Z1 calls ZOL to check if this new sensor data block would overlap any sensor data blocks already defined. ZOL checks the sensor data block of every sensor which is defined to insure that no overlap occurs. If the sensor data blocks would overlap, Z1 aborts, and the DAS prompts for a new command, as before. Otherwise, Z1 requests the upper tolerance ("7-UT= ") and the lower tolerance ("8-LT= "). Each individual character typed on the Teletype to set these first eight parameters is read in by HCKIN, which Z1 calls when each character is typed. For the next two parameters, the multiplier ("9-MULT= ") and the offset ("10-OFFS= "), Z1 is again called when each character is typed on the Teletype, but then Z1 calls MASIN to input each of these floating point numbers. After both of these parameters are fully read in, Z1 requests the label ("11-LABEL= ") and the units ("12-UNITS= ") for the sensor. Again, Z1 is called as each character is typed on the

Teletype and stores the character in a buffer.

After the eight unit characters are entered, Z1 asks if it should add the sensor by printing "OK?". A similar sequence of instructions is used here that the V command uses. However, for the X command, the operator may enter a number between 1 and 12, followed by a space, as the response. Z1 is called as each character is typed, and when it detects the space following the final digit it checks if the number entered is within the range of 1 thru 12. If it is, Z1 sets up a block output instruction to send a carriage return, line feed, and rubouts to the Teletype. Z1 then enables TTY output interrupts and returns to the interrupted program.

The TTY output end-of-block interrupt calls Z1, which then sets up a block output instruction to print the request for the parameter whose number was entered. Z1 also sets up a block input instruction to read one character from the Teletype. Z1 then enables TTY input and output interrupts, the 3 minute time-out in RTCLK, and returns thru TTYOUT to the interrupted program.

Z1 is again called as each character of the parameter's value is typed by the operator. When the complete value is entered, Z1 again asks if it should now add the sensor to the DAS. The operator may also abort the X command by typing "N" in response to the "OK?" message, as described for the V command. However, if "Y" is typed as the response, Z1 then calls Z2 to add the sensor.

Z2 begins executing and first gains control of the disk controller as described for the H command. When Z2 does have control, it constructs the sensor definition block from the parameters which were read in via Z1. Z2 then writes the sensor definition block onto both disks. Z2 next writes the new sensor's data block onto all

eight sections of the disks. All eight copies of the sensor data block are filled with missing-data flags at this time, and the time of the most recent sample, contained in the first word of the sensor data block is set to midnight. Z2 writes the sensor data block to all eight sections in case one or more of the sections contains copied sensor data blocks which the DAS is waiting to transfer, via the Dump Mode, to the 6500 Computer system.

As soon as all disk operations are completed, Z2 writes the correct sensor ID word and sensor disk address word into core memory. This completes the addition of a sensor. Z2 then checks DSDREQ to see if the Dump Mode has requested the disk. If so, Z2 executes a jump and store instruction to the address contained at location DSDADR. When the Dump Mode finishes with the disk, it returns control to Z2, which then returns to Z1. Z1 prints the "ADDED" message and calls PROMPT to request a new command, as detailed for the V command.

4.11.12 Y Command

The Y command removes an individual sensor. It begins executing by entering the VWXY routine thru the Y entry point. VWXY then executes the same sequence of instructions that it does for the V command. In this case, however, the message printed is "REMOVE SENSOR", and ZRFLG is set for a different entry point into Z1.

The remainder of the Y command operates in an analagous fashion to the W command, which removes an entire Remote Station from the DAS. For the Y command, however, the sensor number as well as the station number is requested by Z1 via the same sequence of instructions used by the V command. After the station and sensor number are

entered, Z1 verifies that the sensor is already defined. If it is not, Z1 aborts, and PROMPT then requests a new command, as before. Otherwise, Z1 checks if the Dump Mode or the Automatic Acquisition Mode is processing data from the Remove Station at which the sensor is located. If so, Z1 aborts the Y command, and PROMPT again requests a new command. The Y command may be attempted again in a few minutes.

When Y is called and neither the Dump Mode nor the Automatic Acquisition Mode are processing data from the requested station, Z1 removes the sensor by writing :FFFF into its disk address word in core memory. Z1 then proceeds to print the "DELETED" message and calls PROMPT, as explained for the W command.

CHAPTER V

CONCLUSION

5.1 Objectives

This research has been directed towards developing a Data Acquisition System (DAS) for a specific large-scale, distributed, physical system: the Water Quality Management Facility of the Institute of Water Research at Michigan State University. The DAS was to consist of a centrally-located data logger (Central Station) and several widely-distributed groups of sensors, connected via communication channels. Data from the sensors was to be automatically recorded at fixed intervals of time. Once a day this data was to be transferred to a main computer via a dial-up telephone line. Additionally, an operator was to be able to easily obtain the current value of any sensor, obtain the data automatically gathered from a sensor during the previous 24 hours, add or remove a sensor, and check the overall status of the DAS. Operator access to the DAS was to be possible both from the location of the data logger and from the Water Quality Management Facility. Finally, the DAS was to be able to correctly restart itself after a power failure had occurred.

5.2 Accomplishments

A Data Acquisition System (DAS) which fulfills all the design objectives has been designed, constructed, and tested. A mini-computer was selected to implement the function of the data logger

(Central Station). The communication channels which link the data logger to the remotely-located groups of sensors were realized as dial-up telephone lines. Each group of sensors is interfaced to a separate telephone line via a Remote Station, which samples a sensor and transmits the data to the Central Station when the Central Station sends it the correct command.

A Real-Time Operating System (RTOS) has been written in assembly language for the minicomputer. The RTOS controls the entire data acquisition process. It is designed so that several different operations may be in progress at the same time: sensors can be automatically sampled while a block of sensor data is being transferred to the main computer and an operator is using one of the Teletypes. The RTOS requires 16K of core memory in the minicomputer.

The Central Station sends the data it has automatically gathered from the sensors to the main campus computer. This data is transferred via a dial-up 300 baud telephone line every weekday morning at 9:00. A compact format for sending the data to the main computer has been developed. Checksum characters are transmitted along with the sensor data to provide for error detection. Transmission errors cause the Central Station to retransmit the data to insure that only valid data is entered into the main computer's data base.

Operator access to and control of the DAS is provided by two Teletypes. A controller was designed, constructed, and tested which interfaces both Teletypes to the minicomputer. The Teletype at the Water Quality Management Facility is connected to the minicomputer via a dedicated telephone line and two modem/controllers. An easy-to-use series of commands has been implemented via the RTOS which

allows an operator to: add and remove sensors and Remote Stations, obtain the current value of any sensor, obtain a listing of the data automatically gathered from a sensor during the previous 24 hours, and obtain the status of the DAS. The commands have been designed so that the operator is not forced to remember a complex set of rules to communicate with the DAS. Also, the RTOS allows the operator to correct typing errors by entering the back-arrow (←). This considerably simplifies operator interaction with the DAS.

A battery-operated Real-Time Clock has been designed, constructed, and tested. This Clock is interfaced to the minicomputer and permits the RTOS to determine the correct time after a power failure has occurred. The RTOS then automatically restarts the DAS by writing missing-data flags into the data blocks of those sensors which should have been sampled when the power was off. Thus only a minimal amount of sensor data is lost.

At the present time, the Central Station and one Remote Station have been constructed. Final testing of the DAS in the laboratory is currently in progress. The cost of the equipment comprising the Central Station is \$18,215.00, while the cost of the single Remote Station is \$1955.00. (See Table 5.1) Approximately 300 hours were spent assembling and testing the hardware at the Central Station; the Remote Station would require approximately 120 hours to duplicate. All of the equipment at the Central Station, except the Bell System Data Couplers, the Teletypes, and a small (24" x 24" x 6") cabinet for the Remote Teletype controller is installed in a 6' high standard 19" rack cabinet.

Table 5.1 Cost Breakdown

Central Station

Minicomputer w/16K core memory	\$ 4795.00
Modem/Calling Units (2) (including interface cards)	3600.00
Dual Floppy Disk Unit (including interface card)	4300.00
Teletypes (2)	3400.00
Custom Controller	325.00
Display Controller and Display Panel	210.00
Local Modem/Controller	275.00
Battery Real-Time Clock	220.00
Remote Modem/Controller	290.00
Power Supplies	300.00
Cabinets	300.00
Miscellaneous	<u>200.00</u>
TOTAL COST	\$18215.00

Remote Station (8)

Transmitter Card	\$ 380.00
Receiver Card	330.00
Digital Control Multiplexer Card	30.00
Analog Signal Multiplexer Card	350.00
Modem Card	90.00
Power Supplies	300.00
Cabinet	100.00
Miscellaneous	<u>375.00</u>
TOTAL COST	\$1955.00

5.3 Overall Project Evaluation

The Data Acquisition System described in this thesis effectively fulfills the data gathering requirements at the Water Quality Management Facility. All of the design objectives have been satisfied.

Using a minicomputer for the data logger and dial-up telephone lines for the communication channels linking the data logger (Central Station) and the groups of sensors have proved to be a cost-effective solution to a difficult data-gathering problem. However, changes can be made in the design of the Remote Stations, which interface the groups of sensors to dial-up telephone lines. The Remote Stations were constructed utilizing several SERDEX circuit modules, which are manufactured by Analog Devices Inc. (9,10,11) Today it would be less expensive to design and construct a Remote Station utilizing a microprocessor rather than the SERDEX modules. This change would also allow faster, and therefore more efficient, exchange of data between the Central Station and the Remote Stations.

The two Teletypes provide adequate operator access to the DAS. However, full sensor history listings require several minutes to be printed. Operator access could be improved by replacing the Teletypes by CRT terminals and line printers. This improvement would allow long data listings to be obtained much faster. Operator access could be expanded by interfacing additional dial-up telephone lines to the Central Station. The RTOS could then be modified to allow a user to call the DAS to obtain current sensor data.

The RTOS currently restarts the DAS after a power failure by adding missing-data flags to the correct sensor data blocks. An Uninterruptable Power Supply could be used to power the Central

Station; this would permit it to sample sensors even when commercial power was unavailable. This would be a desirable addition if it was critical to always record data from the sensors.

The DAS now derives the time and date from the power line frequency, via the Real-Time Clock option installed in the minicomputer. A receiver tuned to the National Bureau of Standards time station WWVB could be interfaced to the minicomputer. This time signal has much better long-term accuracy than the power line frequency. The RTOS could then be modified to check the WWVB time once a day and correct the DAS time if necessary. With this addition the DAS would always have the correct time.

APPENDICES

APPENDIX A

BLOCK DIAGRAM COMPOSITION

This appendix lists the integrated circuits which are contained in each functional block on the block diagrams in Chapter 2. Tables are presented for the Custom Controller, the Local Modem/Controller, the Display Controller, the Battery Real-Time Clock, and the Remote Modem/Controller.

Table A.1 Custom Controller Block Diagram IC Composition

<u>Function Block Name</u>	<u>Included Integrated Circuits</u>
Decoders and Latches	1/2 13, 1/2 14, 3/4 25, 26, 27, 28, 29, 30, 1/4 37, 1/2 39, 42, 56, 1/3 72
Device Address Decoder	2/3 6, 1/4 13, 1/3 17, 18
Line Drivers A	2, 3, 4, 5, 1/4 13
Line Drivers B	1/3 38, 1/4 39, 1/2 49, 1/2 50, 1/2 51, 1/2 52, 1/2 61, 1/2 66, 1/2 67
Line Drivers C	1/3 38, 1/4 39, 1/2 49, 1/2 50, 1/2 51, 1/2 52, 1/2 61, 1/2 66
Line Drivers D	40, 1/6 41, 2/3 53, 54
Line Drivers E	1/6 41
Line Receivers A	64
Line Receivers B	15, 16, 1/3 17
Line Receivers C	1, 1/3 6, 1/6 17
Line Receivers D	1/6 17, 1/3 38
Sense Selector	1/4 14, 1/4 37, 1/2 72

Table A.2 Local Modem/Controller Block Diagram IC Composition

<u>Function Block Name</u>	<u>Included Integrated Circuits</u>
Auto Echo	1/4 20
Clock	13
Debouncer	4, 8, 1/3 9, 1/6 11
Decoder and Latch	1/2 12, 15, 1/3 16, 17, 1/2 18, 19
Drivers A	1/2 22
Drivers B	1/2 22
Input Selector	1/2 10, 1/6 11
Line Driver	1/3 9, 1/6 21
Line Receivers A	1/3 11, 1/3 12
Line Receiver B	1/6 12
Manual Power Switch	1/4 20
Monostable	1/2 23
Originate Modem	1, 2, 3, 5, 6, 7
Output Selector	1/4 10
Printer Multiplexer and Enable Delay	1/6 16, 1/2 18, 1/4 20, 1/2 23, 3/4 25
Send Constant Space	1/3 9
Teletype Keyboard Interface	1/3 11
Teletype Printer Interface	1/6 21
Teletype Relay Driver	1/6 16, 1/6 21
UART Receiver	1/2 14
UART Transmitter	1/2 14

Table A.3 Display Controller Block Diagram IC Composition

<u>Function Block Name</u>	<u>Included Integrated Circuits</u>
BCD to 7-segment Converters	15, 18, 20, 23, 26, 29, 31, 34, 36
Latches and Drivers	2, 6, 7, 8, 1/2 9, 2/3 10, 11, 3/4 12, 16, 17, 19, 21, 22, 24, 25, 27, 28, 30, 32, 33, 35, 37
Line Receivers A	3, 4
Line Receivers B	5/6 5

Table A.4 Battery Real-Time Clock Block Diagram IC Composition

<u>Function Block Name</u>	<u>Included Integrated Circuits</u>
Counters	1/2 3, 1/2 4, 2/3 5, 6, 1/2 7, 1/3 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1/4 19, 1/3 45, 1/6 48, 1/6 49, 1/2 50
Latches	1/6 49, 51
Level Converters and Latches	23, 24, 25, 26, 1/3 27, 28, 29, 30, 31, 32, 41, 42, 1/2 50
Line Drivers	1/3 45, 46, 47
Line Receivers	5/6 48, 1/6 49
Line Receivers and Level Converters	18, 3/4 19, 34, 43, 1/3 45
Multiplexers	37, 38, 39, 40
Oscillator	1, 2, 1/2 3, 1/4 4, 1/4 7, 1/6 27, 1/6 33, 1/2 52

Table A.5 Remote Modem/Controller Block Diagram IC Composition

<u>Function Block Name</u>	<u>Included Integrated Circuits</u>
Answer Modem	1, 2, 3, 5, 6, 7
Auto Echo and Manual Override	1/4 15, 1/4 17, 1/4 18
Clock	10
Debouncer	4, 12
Decoder and Drivers	1/2 9, 1/4 13, 14, 1/3 19, 1/2 22, 23
Decoder and Latches	8, 3/4 15, 16, 1/2 17, 1/4 18, 1/2 19, 1/2 20, 25
Drivers A	1/2 24
Drivers B	1/2 22
Manual Power Switch	1/4 13, 1/4 18
Printer Multiplexer and Enable Delay	1/4 17, 1/2 20, 27, 1/2 28
Reset Circuit	1/2 9, 26
Teletype Keyboard Interface	1/6 19
Teletype Printer Interface	1/6 21
Teletype Relay Driver	1/6 21
Transmit Multiplexer	1/4 13, 1/4 18
UART Receiver	1/2 11
UART Transmitter	1/2 11

APPENDIX B

DISK ERROR CODES

An error condition in either floppy disk drive causes the DAS to terminate all processing. The cause of the error must then be determined and manually corrected. To help determine the cause of the error, the disk status word is displayed in the console data register on the front panel of the minicomputer. Each bit, when set, indicates the following conditions:

Bit 0	selected unit not ready
Bit 1	DMA rate error
Bit 2	end of media
Bit 3	attempt to write on write-protected unit
Bit 4	no ID compare
Bit 5	ID CRC error
Bit 6	data CRC error
Bit 7	sync error
Bit 8	non-deleted data encountered
Bit 9	deleted data encountered
Bit 10	done, operations complete
Bit 11	cylinder zero sensed
Bit 12	illegal parameter
Bit 13	controller not present
Bit 14	hardware error (timeout)
Bit 15	write protected unit

The OV lite is set if disk unit 1 caused the error, and reset if unit 2 caused the error.

APPENDIX C
BASE PAGE INTERRUPT LOCATIONS

The interrupt structure of the minicomputer is utilized by both modem/calling units, the Teletypes, the real-time clock, and the power line monitor. The following list shows the location in the minicomputer's memory where each particular interrupt is vectored. A single instruction at that address is executed when the corresponding interrupt occurs.

:00	Power Up
:02	Teletype Output
:06	Teletype Output End-Of-Block
:18	Real-Time Clock
:1A	Real-Time Clock Sync
:1C	Power Down
:1E	Console/Trap
:22	Teletype Input
:26	Teletype Input End-Of-Block
:60	ACU #1 - Abandon Call and Retry (ACU)
:62	ACU #1 - Data Set Status (DSS)
:64	ACU #1 - Present Next Digit (PND)
:68	ACU #2 - ACR
:6A	ACU #2 - DSS
:6C	ACU #2 - PND
:80	AMC #1 - Receive
:84	AMC #1 - Transmit
:88	AMC #2 - Receive
:8C	AMC #2 - Transmit
:92	AMC #1 - Exception
:94	AMC #1 - Receive End-Of-Block

:96 AMC #1 - Transmit End-Of-Block

:9A AMC #2 - Exception
:9C AMC #2 - Receive End-Of-Block
:9E AMC #2 - Transmit End-Of-Block

LIST OF REFERENCES

REFERENCES

1. Utilization of Natural Ecosystems for Wastewater Renovation-Final Report, prepared by the Institute of Water Research, Michigan State University (March, 1976)
2. Floppy Disk Subsystem User's Manual, prepared by Computer Automation, Inc. (August, 1975), p. G-4
3. Installation Instructions-Standalone 300 BPS Modem/Dialer Combination, prepared by the Vadic Corporation, (December, 1975)
4. Bell System Technical Reference-Data Couplers CBS and CBT for Automatic Terminals, prepared by American Telephone and Telegraph Company, (May, 1974)
5. Low-Speed Modem Fundamentals (AN-731), prepared by Garth Nash, Computer Systems Engineering, Motorola Semiconductor Products, Inc., (August, 1974)
6. MC6860L 0-600 BPS Digital Modem-Advance Information, prepared by Motorola Semiconductor Products, Inc.
7. NAKED MINI/ALPHA LSI Series Computer Handbook, prepared by Computer Automation, Inc., (July, 1974), p. 6-22
8. Krause, Gary S., A Minicomputer-Based Remote Data Acquisition System, Michigan State University, (1976)
9. SERDEX User's Guide-Serial Receiver SRX1005 and SCL1006, prepared by Analog Devices, Inc.
10. SERDEX User's Guide-Serial Multiplexer SMX1004/SMC1007 and SCL1006, prepared by Analog Devices, Inc.
11. SERDEX User's Guide-Serial Transmitter STX1003 and SCL1006, prepared by Analog Devices, Inc.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 03145 8106