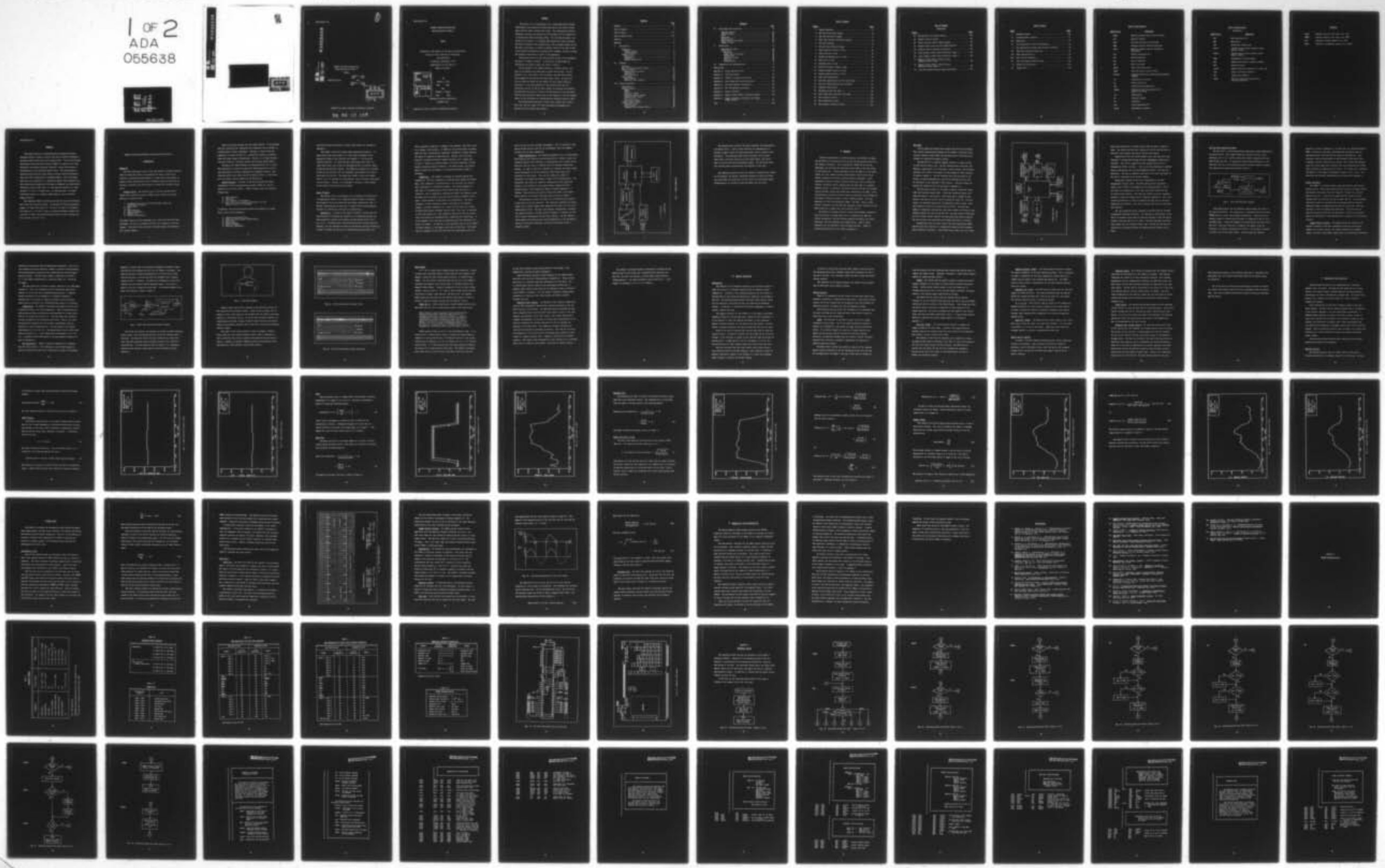
AD-A055 638 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 6/16
AIRCREW INFLIGHT PHYSIOLOGICAL DATA ACQUISITION SYSTEM II.(U)
DEC 77 J G JOLDA, S J WANZEK

UNCLASSIFIED AFIT/GE/EE-77-21 NL

1 OF 2
ADA
055638



AFIT/GE/EE/77-21

①

AIRCREW INFLIGHT PHYSIOLOGICAL DATA ACQUISITION SYSTEM II

THESIS

Joseph G. Jolda
Capt USAF

AFIT/GE/EE/77-21

and

Stephen J. Wanzek
Capt USAF

DDC
RECEIVED
JUN 22 1978
E

Approved for public release; distribution unlimited

78 06 13 189

AFIT/GE/EE/77-21

AIRCREW INFLIGHT PHYSIOLOGICAL
DATA ACQUISITION SYSTEM II

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Joseph G. Jolda
Capt USAF
and
Stephen J. Wanzek
Capt USAF
Graduate Electrical Engineering
December 1977

Approved for public release; distribution unlimited.

Preface

This thesis is an investigation into a second generation Inflight Physiological Data Acquisition System sponsored by the School of Aerospace Medicine (SAM) at Brooks AFB, Texas. The investigation entailed designing, building, and testing both the hardware and the software for a microprocessor-based prototype system. The resulting prototype, presented in this report, is a complete data acquisition system including the sensor interfaces, the microprocessor, and a permanent memory device. The body of the report is written in general terms for the user at SAM; a more detailed description including circuit diagrams, program listings, and technical discussions is given in the appendices.

This project would not have been possible without the encouragement and help of a number of people. We would like to acknowledge the individuals who helped to make this thesis a reality.

We are indebted to Dr. Mathew Kabrisky, our thesis advisor, who gave us the guidance and enthusiasm to pursue this project. We are grateful to Dr. Gary Lamont, Capt Mike Weber, and Capt Chuck Cornell whose suggestions and assistance helped immeasurably. We would also like to thank Bob Durham and Dan Zambon for their superb technical assistance. We are also grateful to the personnel from the Crew Technology Division at SAM for their support in obtaining the hardware. Our gratitude also goes to Al Haun of Analog Devices for his assistance with the data acquisition module and to Jack Capehart at the ASD computer center for his assistance in transferring the operating system to PROM.

Our deepest gratitude goes to Steve's wife, Cindy, and to Greg's wife, Sue, and son, Jason, for their continuing encouragement and patience while we pursued this project.

Contents

Contents

List of Figures

List of Figures

(Continued)

List of Tables

List of Abbreviations

Abbreviation	Definition
ABSPR	Absolute pressure sensor service routine
A/D	Analog to digital
AFIT	Air Force Institute of Technology
AMRL	Aerospace Medical Research Laboratory
ASCII	American Standard Code for Information Interchange
CPU	Central processing unit
ECG	Electrocardiogram
EOC	End of conversion
EOI	End of interrupt
ERROR	Error correcting routine
FLWRT	Flow rate sensor service routine
FORTRAN	(FORmula TRANslation) Engineering programming language
G's	Acceleration of Gravity
Hz	Hertz (cycles/sec)
ICW	Initialization command word
IFPDAS	Aircrew Inflight Physiological Data Acquisition System
I/O	Input/output
IR	Interrupt request
IS	In-service
LSB	Least significant bit
mm Hg	Millimeters of Mercury

List of Abbreviations

(Continued)

Abbreviation	Definition
MSB	Most significant bit
MUX	Multiplexer
OCW	Operational command word
PO2IN	Inspired oxygen partial pressure sensor service routine
PO2OUT	Expired oxygen partial pressure sensor service routine
PROM	Programmable read-only memory
RAM	Random access memory; read/write memory
ROM	Read-only memory
R-wave	Highest amplitude component of a normal ECG
SAM	School of Aerospace Medicine
SBC	Single board computer
USART	Universal synchronous/asynchronous receiver/transmitter

Notation

(name)	Signifies positive true logic (e.g., EOC)
($\overline{\text{name}}$)	Signifies negative true logic (e.g., $\overline{\text{EOC}}$)
xxxx	Signifies a decimal number (e.g., 3184)
xxxxH	Signifies a hexadecimal number (e.g., 0C70H)

Abstract

This paper discusses a second generation microprocessor-based prototype system to acquire, analyze, and store selected environmental and physiological data from a pilot during flight. The Aircrew Inflight Physiological Data Acquisition System (IFPDAS) II consists of an input multiplexer and analog-to-digital converter, a heart rate detector, a microprocessor, and a permanent memory device. The microprocessor's operating system monitors eight sensors, extracts desired information, and stores these reduced data in permanent memory. After the flight, these data are transferred to a land-based computer which completes the data processing and graphs the following environmental and physiological information versus flight time: (1) cabin absolute pressure, (2) cabin altitude, (3) Z-G's, (4) heart rate, (5) breathing rate, (6) minute ventilation volume, (7) inspired oxygen quantity, and (8) expired oxygen quantity.

The completed IFPDAS II prototype provides the desired information well within the required accuracy. It provides the following parameter ranges: (1) heart rate from 53 ± .1 to 225 ± 2.2 b/min, (2) breathing rate from 4.7 ± .1 to 50 ± 1 b/min, (3) minute ventilation volume from 0 to 100 ± 2 l/min, (4) absolute pressure from 0 to 760 ± 2 mm Hg, and (5) G's from -3 to +12 ± .1 G.

AIRCREW INFLIGHT PHYSIOLOGICAL DATA ACQUISITION SYSTEM II

I Introduction

Background

The Crew Technology Division of the USAF School of Aerospace Medicine (SAM) at Brooks AFB, Texas, has recognized the need to relate pilot activity to physiological measurements, to apply these relationships to predict aircrew effectiveness, and to formulate equipment design and use criteria to optimize that effectiveness in present and projected flying roles.

Current System. SAM currently has an "Aircrew Inflight Physiological Data Acquisition System" (IFPDAS) which records seven analog functions on cassette tape:

1. A standard time code (to correlate flight events and physiological effects);
2. Pilot voice;
3. ECG;
4. Cabin pressure;
5. Oxygen consumption;
6. Expired flow;
7. Vertical acceleration.

The IFPDAS consists of two subsystems: one to sense and record the data (inflight), the other to reproduce the data (on the ground - after the flight). This data is then converted to digital signals and processed by a digital computer.

There are several problems with the current IFPDAS. It was designed and built using discrete components and is therefore not as reliable as a system based on modern components. Secondly, it doesn't have the capability to acquire triaxial G's, inspired flow volume, or separate input and output oxygen concentrations. Finally, it is highly specialized and, therefore, inflexible without costly design modifications.

Currently, the IFPDAS hardware is being modified by the U.S. Navy. This modification includes two additional analog functions as well as some replacement of discrete components by integrated circuits. The resulting modification should be more reliable; however, it still won't have the complete desired capability and flexibility.

System Standards. Personnel at SAM have projected the design requirements for the second generation system (IFPDAS II), due for production in the early 1980's. IFPDAS II must provide the following primary data:

1. Cabin pressure;
2. Time code;
3. G's (triaxial, if available);
4. Voice (this can be acquired separately if an all-digital system can be designed).

In addition, this new system should provide the capability to assess three or more of the following:

1. Inspired and expired flow;
2. Input and output oxygen concentrations;
3. Heart rate;
4. Mask pressure;
5. Garment pressure;
6. Body core temperature.

The desired range and accuracy for each of the sensors is included in Appendix A.

The IFPDAS II must meet several other system specifications. It must accept probe inputs from 0 to 5 volts which correspond to the appropriate range of each function (for example, 0 - 760 mm Hg for absolute pressure). It should provide a data acquisition time of at least three hours, and it must be time-synchronized to correlate flight events and physiological effects. The IFPDAS was designed to be carried in the pilot's survival vest and, therefore, the original size restrictions must still be met. This means that IFPDAS II must be no larger than 2" x 5" x 9" and must be self-contained, with no external connections to the aircraft. Finally, it is desirable to present a visual display of the status of the device and its probes.

Scope of Thesis

The purpose of this investigation was to determine the feasibility of implementing IFPDAS II as a completely digital system to eliminate the mechanical drives and reduce the post-flight computations; and, if digital implementation was practical, to update the existing equipment, increase its reliability, and extend its capabilities.

Feasibility. A search of the current literature revealed that the microprocessor is extending the capabilities of monitoring systems and data acquisition systems in the medical and engineering fields (Ref 1, 2, 3, & 4). A microprocessor controlled system offers several excellent features, the most important of which are flexibility and high reliability. In order to modify the function of a microprocessor-based system, all that is generally required is a change in the software, often with little or no change in the hardware. In addition, the microprocessor can manipulate the data, extracting the significant information, thereby reducing the amount of permanent storage required. Finally, since the microprocessor incorporates numerous digital functions onto a single unit, or chip, it replaces an enormous number of discrete components, thereby greatly increasing system reliability. For these reasons, the investigation includes the development of a microprocessor-based IFPDAS II prototype.

Assumptions. This IFPDAS II prototype was designed making four assumptions. The first assumption is that the probes supply the desired data. This assumption is required since it is not within the scope of this investigation to redesign the probes. The second assumption is that there is a maximum period of four hours during which data is collected. This is required to establish the memory size needed to store the data, and is justified since the current system is limited to four hours - which personnel at SAM found to be satisfactory. The third assumption is that continuous storage of the data is not required. Instead, a periodic technique (i.e., every 10 to 30 secs) or a "store-on-significant-change" technique could be used. This, too, is necessary to limit the memory size. This assumption should not limit the usefulness of the data since unchanging data is generally not interesting. It is the changes in the data that are important, and both techniques will detect the changes. In addition, it will generally take 10 to 30 seconds to detect changes in, for example, heart rate or flow rate. The fourth and last assumption is that the power and size requirements would not have to be met for this prototype development. This is required so that proven systems could be used for the development, with less emphasis on their size or power consumption.

System Configuration. The minimum configuration for a microprocessor-based digital data acquisition system would have to include a sensor, an analog-to-digital (A/D) interface, a microprocessor, and a memory device. When more than one sensor is required, the A/D interface becomes more complex. In order to keep this interface to a minimum, the sensor inputs can be multiplexed to one A/D converter, rather than using an A/D converter for each sensor. This not only reduces the number of converters, but reduces the number of inputs to the microprocessor. The multiplexer and single A/D converter configuration allows additional sensors to be interfaced to the system without changing the basic system hardware, while keeping the number of system components to a minimum. These concepts were used to design the IFPDAS II prototype, which consists of six major functional units as shown in Figure 1.

The multiplexer selects one from up to 16 different probe signals as the input to the A/D converter. This converter transforms the analog signal into a digital number which represents the signal for use by the microprocessor (CPU). The heart rate detector (implemented in hardware for required accuracy - see ECG section of Chapter II) supplies the CPU with a number representing the heart beat interval. The CPU combines these numbers with previous data (stored temporarily in the CPU memory), extracts the desired information, and stores the desired result in permanent memory.

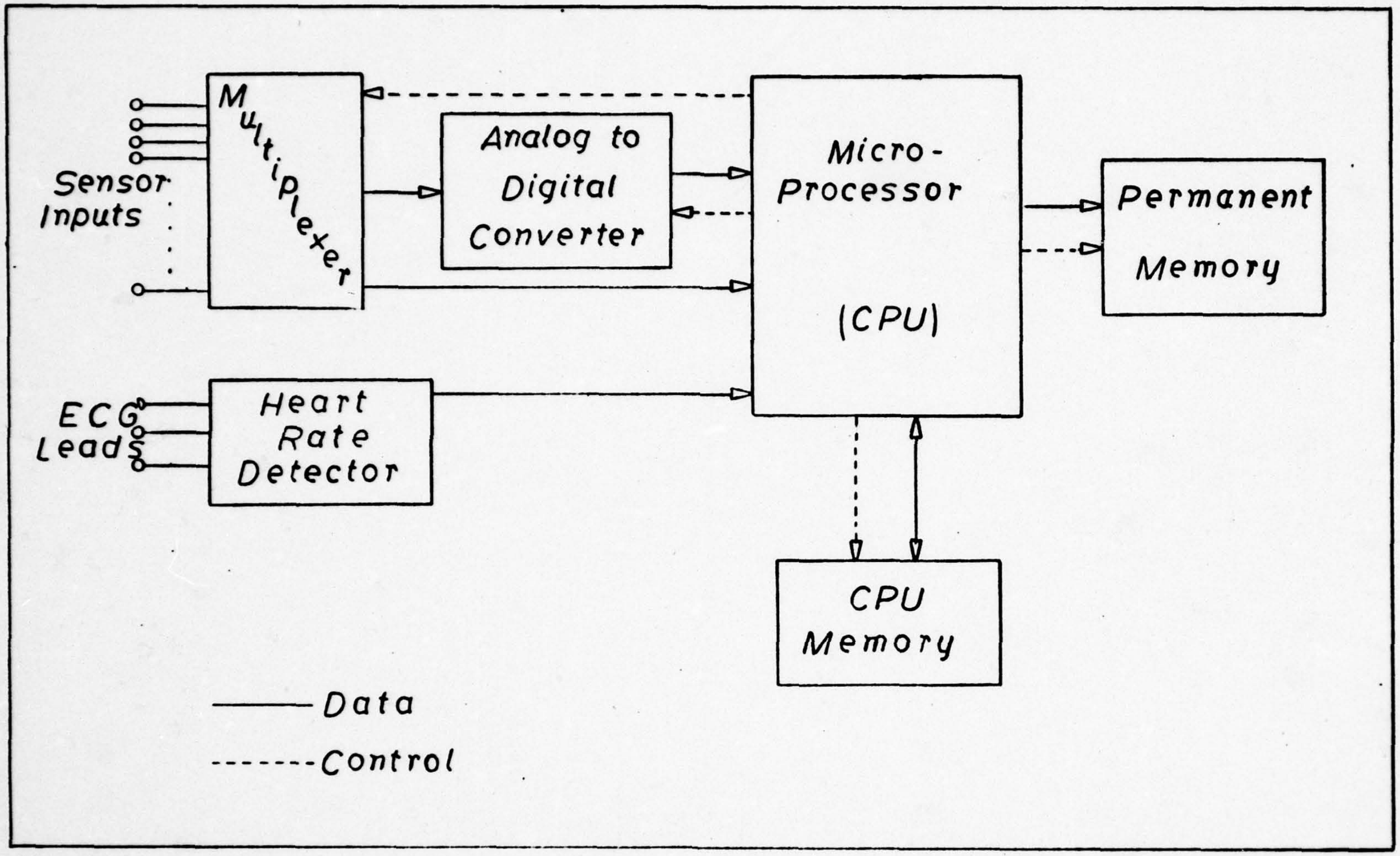


Fig. 1. System Components

The microprocessor performs two major functions: data manipulation and system control. Both of these functions are implemented by a software program, called the operating system, which is stored in the CPU memory. The operating system directs selection of the proper input probe, initiates conversion of the analog signal, and routes data to and from both CPU and permanent memory. The operating system also ensures proper interpretation and reduction of the data.

The remaining chapters discuss the IFPDAS II prototype with respect to its hardware, the general algorithms designed to acquire the data, the post-flight data reduction, and the tests of the prototype system. Recommendations for further system development are also given.

II Hardware

Several microprocessors (including the Z80, the TMS9900, the 8080, and the 6800) were considered for use as the central processing unit for the IFPDAS II prototype. Intel Corporation's 8080A CPU has several advantages over the other microprocessors which make it the best choice for the system CPU. A major advantage is that the 8080 is a well established, low cost, highly reliable microprocessor (Ref 5:44-45) which readily integrates with Intel's general purpose peripherals. These peripherals provide a variety of special functions (such as timing and external interfaces) which, together with the 8080, make up a complete computer system. In addition, the devices selected for the A/D interface and the permanent memory had already been interfaced to the 8080. Other considerations include the author's previous experience with the 8080 and the support software available on AFIT's computer system. One final advantage is Intel's new generation 8080A - the 8085 - which is 100% software compatible with the 8080 and integrates several functions of the 8080 system onto a single chip (Ref 6:109-113, Ref 7).

The IFPDAS II prototype consists of four major hardware components: Intel's SBC 80/20 (a single board computer containing the 8080 and several peripherals), the DAS 1128 Data Acquisition Module, the sensor interfaces, and the permanent data storage device. Each of these components will be described in the following sections. IFPDAS II prototype characteristics are listed in Appendix A.

SBC 80/20

Intel's 8080-based single board computer SBC 80/20 was purchased. This allowed engineering development of the IFPDAS II prototype using a proven computer system rather than devoting time to fabrication and testing of a specialized computer system.

The SBC 80/20 is a complete computer system on a single 6.75-by-12 inch printed circuit card. The CPU, system control functions, CPU memories, input/output (I/O) interfaces, interval timers, and interrupt controller all reside on the board (a block diagram of these functions is shown in Figure 2). The CPU functions have been discussed in the section on system configuration; each of the peripheral devices will be introduced in the remainder of this section. (Specific design and operational characteristics are discussed in Appendix E.)

The CPU memory consists of two types of memory: read/write memory (RAM) and read-only memory (ROM). Unlike the read/write memory, the read-only memory is non-volatile, which means that a program or data stored on the ROM will not be lost by turning the power off. For this reason, the operating system will be permanently stored in ROM. Read/write memory is used like a scratch pad by the CPU. Previous data samples, intermediate calculations, and event counters are stored on this temporary memory for later use by the CPU. Any data stored in RAM would be lost if power failed; however, the operating system would recognize the loss of power and reinitialize the system when power is restored.

Communication with the input devices (the data acquisition module and the heart rate detector) is accomplished through two 8255 Programmable Peripheral Interfaces. These 8255s receive data over 8 or 16 data

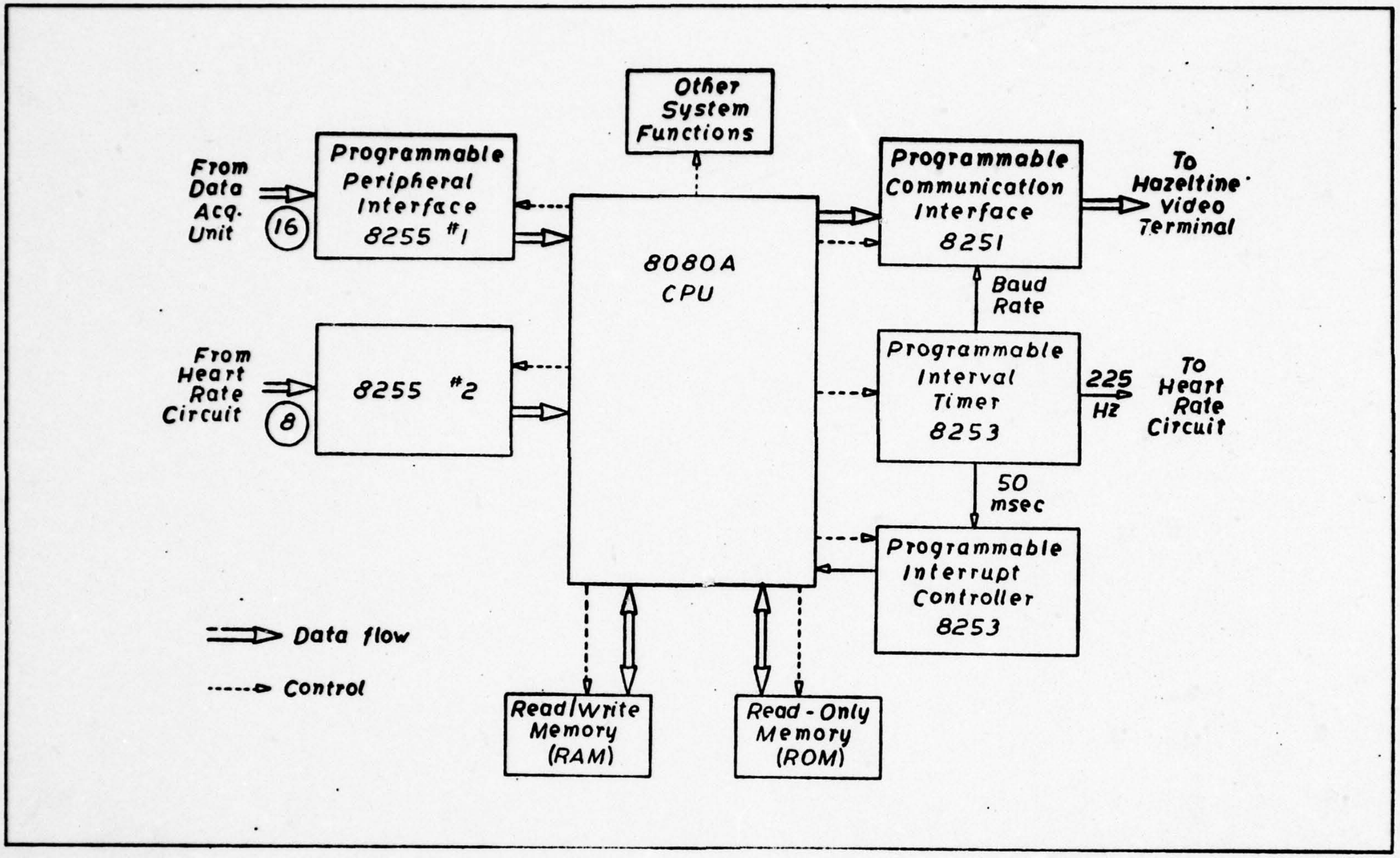


Fig. 2. SBC 80/20 Functional Diagram

lines when informed by an external device that the data is ready for input. The 8255s store the data, indicate to the CPU that the data is available, and transfers the data to the CPU upon request.

Communication with the current memory device (the Hazeltine video terminal) is accomplished through the 8251 Programmable Communication Interface. This device communicates with external devices over a standardized (RS232) interface. (The use of this standardized interface allows communication with any RS232-compatable device - not just the Hazeltine.) The rate of transfer (baud rate or bit rate) and format of the data is controlled by the system software.

Timing and frequency division are accomplished by the 8253 Programmable Interval Timer. The 8253 contains three independent timers, each of which is programmed by the operating system. One of the timers is configured as a real time clock which informs the CPU of every 50 msec interval. The other two timers are used as frequency dividers, slowing the system clock to required frequencies. One of the resulting frequencies is used to establish the baud rate for the 8251 communications interface. The other is used as the clock for the heart rate detector.

The last peripheral device contained on the SBC 80/20 is the 8259 Programmable Interrupt Controller. An interrupt is notification to the CPU by an external device that an event has occurred or that the device requires servicing (Ref 8: Ch 5, 8-33). The 8259 intercepts the interrupt request from the external device, and it informs the CPU that the interrupt has occurred and where the software service routine can be found.

DAS 1128 Data Acquisition Module

The DAS 1128 is a self-contained data acquisition system manufactured by Analog Devices. This compact module was selected for use in the prototype since it is a proven system which readily integrates with the 8080 through the 8255. It contains an analog input multiplexer, an A/D converter, and all of the timing and control circuitry needed to perform the complete data acquisition function (Figure 3).

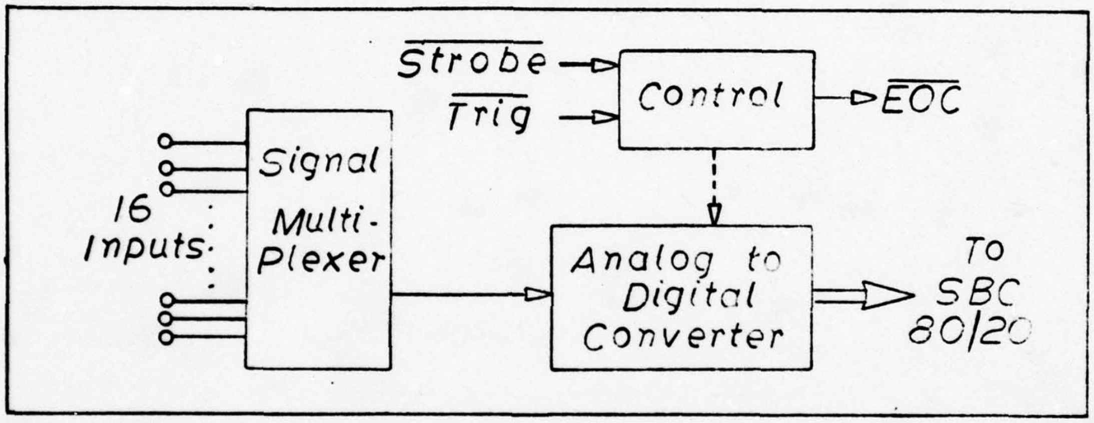


Fig. 3. DAS 1128 Functional Diagram

Seven physiological and environmental analog signals are input to the signal multiplexer. The multiplexer is directed by the CPU (via the $\overline{\text{STROBE}}$ input) to select these signals sequentially. Conversion of the selected signal is started when the module is triggered by the CPU (via the $\overline{\text{TRIG}}$ input). (These CPU commands are transmitted by the 8255 to the DAS 1128.) When the conversion is complete, the output of the A/D converter is a digital representation of the 0 - 5 volt input, accurate to within 10 mv of the input signal. (This provides the required accuracy as listed in Appendix A.) At this time, an "end-of-conversion" ($\overline{\text{EOC}}$) is returned to the 8255, signifying that the digital data and associated sensor identification are ready for transmission. Each conversion requires approximately 25 microseconds, which would allow a maximum of 40,000 conversions per second. This is more than adequate since the current application requires only 1120 conversions per second. (This is based on a sampling rate of 20 Hz. If a faster rate is desired, the IFPDAS II could sample the maximum 16 inputs at over 70 Hz.) The electrical and interface configurations are discussed in Appendix F.

Sensor Interfaces

The IFPDAS II prototype accepts eight physiological and environmental sensors. These probes measure partial pressure of oxygen inhaled and exhaled, respiratory flow rate (expired), triaxial G's, absolute pressure, and ECG. A mask and two oxygen partial pressure sensors (Beckman OM11) were provided by SAM at Brooks AFB, TX. An accelerometer (Statham F-15-340) was obtained from the Aerospace Medical Research Laboratory (AMRL) at Wright-Patterson AFB, OH; and the ECG leads were available in AFIT's Bioengineering Laboratory. (The flow rate and absolute pressure sensors were not available.) Each of the available probes was interfaced to the prototype and is discussed in the following sections.

Oxygen Partial Pressure.

The oxygen sensors are located in the mask/hose assembly in the same configuration used for the original IFPDAS (Ref 9:24-25); however, the oxygen information has changed. IFPDAS I provided a measurement representing the instantaneous difference between the inspired and expired oxygen partial pressures. This is not the information desired; therefore, IFPDAS II provides two measurements, one representing the inspired and one representing the expired oxygen partial pressure. The OM11 sensor output is amplified to provide a 0 - 5 volt signal corresponding to a pressure range of 0 - 760 mm Hg of oxygen.

The OM11 sensor has an 800 msec response time for a 0 to 100% oxygen transition. This is an acceptable time for the present application; however, it is too slow for a breath-by-breath analysis. A complete circuit description of the amplifier is included in Appendix G. Included also is a circuit to reduce the response time to 100 msec, which would be sufficient for a breath-by-breath analysis, if desired.

Acceleration. The IFPDAS II prototype was designed to monitor the acceleration forces in all three dimensions. AMRL at Wright-Patterson AFB has mounted three uni-directional accelerometers on a breast plate (Ref 10) to provide the triaxial G's. Only one sensor was available, so only one amplifier was constructed and tested. However, the other two amplifiers (required for full triaxial G measurement) would be identical to the one actually built. The full range of the Statham accelerometer is -15 to +15 G's. The amplification circuit offsets this range and provides a 0 - 5 volt signal which corresponds to -3 to +12 G's. A complete circuit description of the accelerometer interface is given in Appendix G.

ECG (Heart Rate). IFPDAS I stores the complete ECG on cassette tape for later analysis. This recording is only reliable enough to provide information for heart rate calculation, by hand, on the ground.

Normally, the heart rate is the desired information; therefore a heart rate detector was designed and built for the IFPDAS II prototype. This detector provides a digital representation of the heart rate, along with the provision to record the entire ECG waveform with a separate analog recorder, if desired. The heart rate information is derived by counting the time interval between detected R-waves. This count is passed to the CPU, through the second 8255. A functional diagram of the heart rate detector is given in Figure 4.

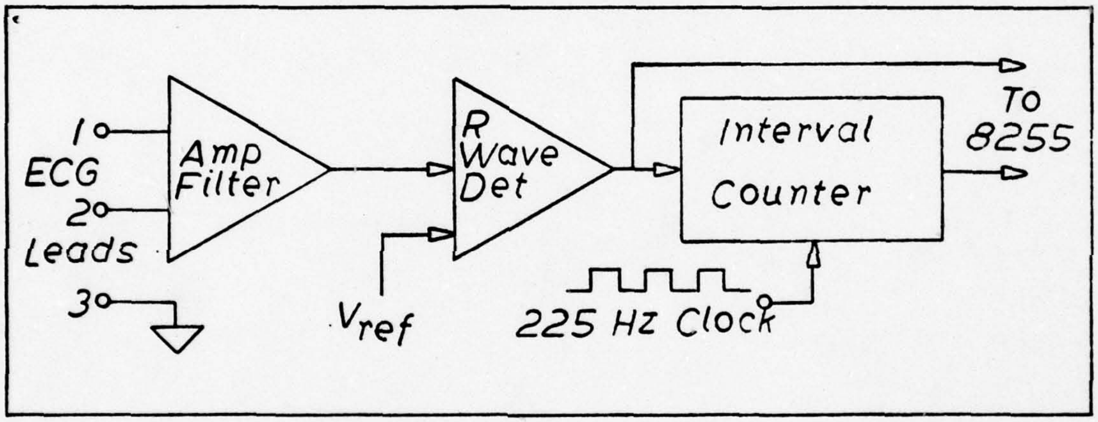


Fig. 4. Heart Rate Detector Functional Diagram

This heart rate detector was designed to provide a maximum amplitude R-wave signal, while eliminating base-line shifts and reducing muscle artifacts. The base-line shifts and muscle artifacts are undesirable since they add extraneous signals and make accurate R-wave detection very difficult. The lead placement shown in Figure 5 is a compromise placement which provides a good QRS-wave least disturbed by muscle artifacts.

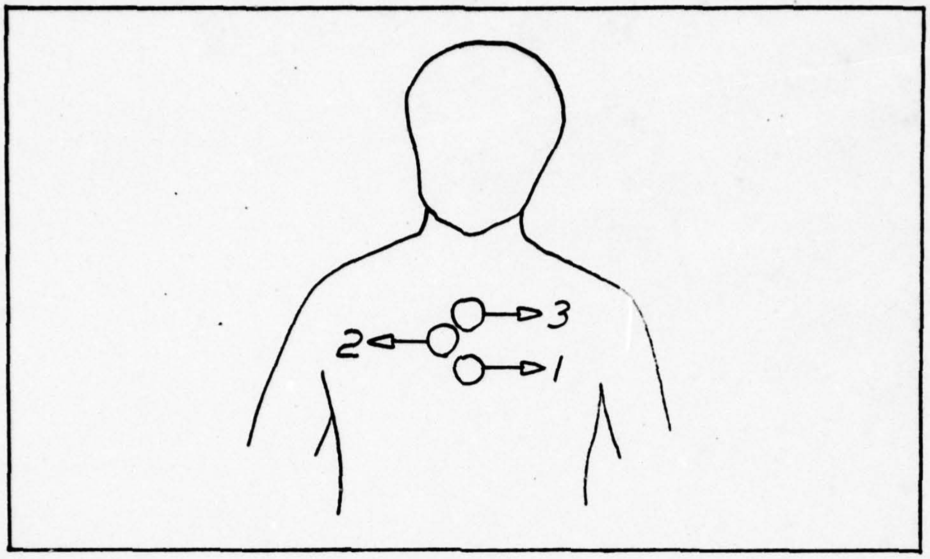


Fig. 5. ECG Lead Placement

Figure 6 shows typical ECG's provided by the amplifier, along with the associated R-wave detector output. Figure 6a was recorded with the subject at rest, while Figure 6b was recorded with the subject exercising heavily on a stationary bicycle. As can be seen from these figures, the R-wave detector provides a highly reliable R-wave indication even under exercise and movement conditions more severe than a pilot would experience during a flight.

The output of the R-wave detector is used to trigger a counter, which counts the time interval between R-waves. When a subsequent R-wave is detected, this count is passed to the 8255 and the new count is started. Appendix G contains a complete electrical description of the ECG amplifier, R-wave detector, and interval counter.

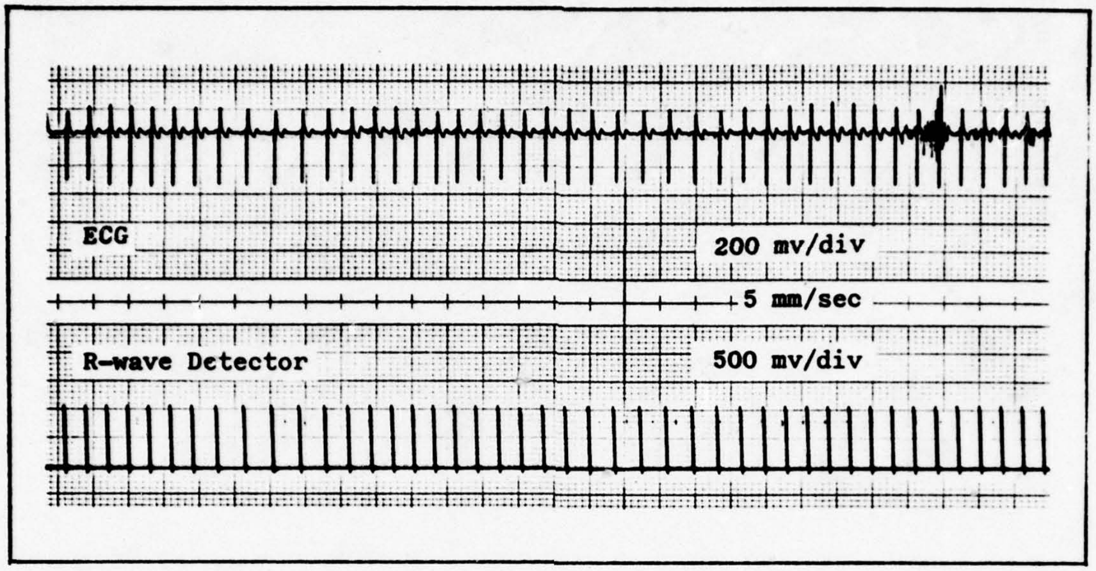


Fig. 6a. ECG & R-wave Detector Output (Rest)

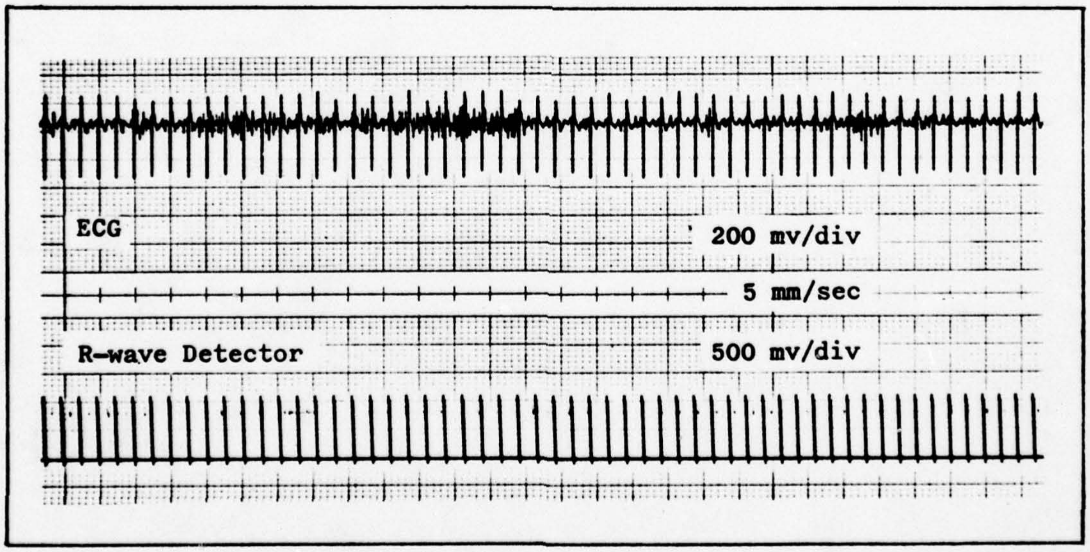


Fig. 6b. ECG & R-wave Detector Output (Excercise)

Data Storage

A variety of current data storage devices were considered. Several of these were eliminated because of their physical size (magnetic core memory), volatility (RAM, charge-coupled devices), or impracticality (PROM). The only devices considered practical for use on the IFPDAS II prototype were magnetic tape (either wafer or standard cassette) and magnetic bubble memory. (Product information for the two tape storage systems is given in Ref 11, 12.) Both of these storage devices provide non-volatile storage, large capacity, and small size. The magnetic bubble memory was selected over the cassette tape systems in order to provide a completely digital system with no mechanical drives.

Bubble Memory.

Bubble memories have several characteristics which make them ideal for use as a mass memory system in this application.

> These attributes result primarily from the semiconductor-like processing used to fabricate bubble devices in conjunction with their non-volatile nature. ... Some of the important features resulting from the semiconductor-like processing include reliability, small size, fast access time (relative to electro-mechanical mass storage devices), low cost, and small incremental capacity. ... By virtue of its non-volatility, the bubble device offers removability, asynchronous access, and low power. (Ref 13:1)

Bubble memory systems are still in the developmental stage. It is expected that a complete memory system, including two bubble memories (capable of storing 184,000 bits of information) and all of the system controls and interfacing, will fit on a single 4.5" x 6" x .75" printed circuit board. It is also expected that the power required to operate the system will be approximately three watts. The IFPDAS II prototype only stores data for 10 msec every 10 sec which would allow the CPU to turn off the memory system during periods of non-storage - thus significantly reducing the power consumption.

Texas Instruments supplied circuit diagrams for the bubble memory controller and functional timing generator (Appendix H). These circuits were wired on a component board and interfaced to the SBC 80/20. Due to limited bubble memory production and procurement difficulties, a complete bubble memory system could not be obtained in time to interface to the IFPDAS II prototype. For this reason a temporary storage method consisting of a Hazeltine 2000 video terminal and digital cassette recorder was used.

Hazeltine Video Terminal. The Hazeltine video terminal communicates with the 8251 Programmable Communication Interface discussed in the SBC 80/20 section of this chapter. The operating system transforms the data into a standard format (seven-bit ASCII code) which is sent to the video terminal and displayed on the video screen. This visual display was a valuable developmental tool because it allowed easy interpretation of the data for validating the software operations as well as checking the accuracy of the stored data. This temporary "storage" technique was used until the system was operating as desired. Then data was collected by displaying it on the video screen and automatically copying it permanently on a digital cassette tape. (Appendix C describes the transfer process.) This digital tape represents the final storage of the "inflight" data, which is ready for "post-flight" conversion and graphic display.

The IFPDAS II prototype hardware configuration includes the SBC 80/20 printed circuit board; and a component board containing the DAS 1128, the heart rate detector, and the bubble memory function timing generator. The analog interfaces that provide the 0 - 5 volt signals are packaged in a 3" x 4" x 5" chassis.

III General Algorithms

Introduction

The operation of the hardware discussed in the previous chapter is under the control of a software program called the operating system (Appendix B). Execution of the operating system by the CPU results in initialization of the system and acquisition, reduction, and storage of the data. The operating system selects the proper input sensor, starts the A/D conversion, and inputs the data for reduction and permanent storage. In addition, it ensures that the proper subprogram, or service routine, receives the data.

The biggest constraint on the IFPDAS II is the amount of available permanent storage for the acquired data. Because of this limitation, it is not possible to store the complete waveforms, or even continuous samples of these waveforms, for the desired four hour period. For this reason, a periodic technique (storing the data every ten sec) was used.

Three of the service routines must sum the input data for each ten second interval. In order to constrain the ten second sum to 16 bits of storage, it was necessary to sample no more than 256 times during the 10 second period. A sampling rate of 20 Hz (200 samples in 10 sec) was selected because the input signals change very slowly (less than 5 Hz).

Data are collected from each input signal every 50 msec using a noise-reducing digital filtering technique. This technique consists of taking 8 consecutive samples of the waveform in .5 msec and averaging these 8 samples to produce the 50 msec reading.

In order to collect data from each input signal at the 20 Hz rate, the operating system sets a hardware timer which interrupts the CPU at the end of 50 msec. This interrupt informs the CPU to start the service routine sequence.

The remainder of the chapter discusses the sensor service routines and the additional system support software.

Service Routines

There is an independent service routine for each input probe since different information is required from each sensor. Each service routine accepts the 50 msec reading from the averaging routine and performs the required operations to prepare the data for permanent storage.

The following sections discuss the information to be extracted from the input waveforms and the algorithm used in each service routine to reduce the data to the desired form.

PO2IN.

The personnel at SAM need to assess the pilot's oxygen consumption during flight. One of the required quantities needed to compute this consumption is the amount of oxygen inspired during each breath. To accurately compute the quantity of oxygen inspired, the oxygen partial pressure and the inspired flow rate are required. Currently, the mask/hose assembly does not contain a sensor to measure inspired flow; therefore, a program to approximate the quantity of inspired oxygen was written.

The PO2IN service routine was written to sum all of the inspired partial pressure readings for the ten second period and store the sum for averaging after the flight. This sum is then used in conjunction with the expired flow rate (assuming equal inspired and expired rates) to compute the oxygen intake. (Appendix I discusses a single breath analysis method for computing oxygen uptake.)

P02OUT. The second required quantity needed to compute the pilot's oxygen consumption is the amount of carbon dioxide produced during each breath. Current carbon dioxide sensors do not lend themselves to inflight applications due to their size and weight; therefore, an approximation of the expired oxygen quantity had to be made.

The P02OUT service routine sums the expired partial pressure readings for the ten second period and stores the sum for the post-flight conversion, which is similar to the P02IN methods. (Even though the P02IN and P02OUT service routines are inaccurate in their analysis of the oxygen quantities, the sensor interfaces and data handlers were successfully exercised and produce approximate values. A single breath analysis method for computing the expired oxygen quantity is included in Appendix I.)

Flow Rate (FLWRT). The third quantity required to compute the oxygen consumption is flow volume. A probe in the oxygen mask/hose assembly measures the expired volume flow rate, which can be integrated with respect to time to obtain flow volume.

The integral of the flow rate waveform can be computed by using a rectangular approximation technique, since there is very little change to the flow rate signal during a 50 msec interval. The FLWRT service routine sums and stores the 200 flow rate readings for subsequent multiplication (after the flight) by the known 50 msec interval to produce the resulting integral.

Absolute Pressure (ABSPR). The final quantity required to compute the oxygen consumption is the cabin absolute pressure. This is necessary in order to compensate the flow rate reading for altitude and also to compute percent oxygen in the inspired and expired air. The ABSPR service routine stores a representation of the cabin absolute pressure every ten seconds.

Breathing Rate (BRRT). The BRRT service routine uses the flow rate sample for breath detection. The routine determines when the exhaled breath has stopped and marks this event as the start of a new breath. The interval between breaths is counted and stored.

X, Y, & Z G's. These service routines search for the maximum and minimum acceleration in each direction during the ten second interval. In order to reduce the effects of transient G impulses, each routine averages eight readings before comparing to the previous minimum and maximum values.

Heart Rate (HEART). The HEART service routine checks to see if the heart rate circuit has input a new heart beat interval. If a new count is available, it is read in and stored. Eight heart beat counts are averaged together to provide a representative heart rate.

System Support Software

In order to form the complete operating system, several additional routines are necessary. These routines initialize the IFPDAS II prototype, correct detectable errors, store the data on the permanent storage device, and provide the additional support required by the service routines.

Power-up Routine. This routine is executed when the hardware detects that power has been applied to the IFPDAS II prototype. This program configures the IFPDAS to its data acquisition function. This includes programming the 8251 communication interface, the 8255 input ports, the 8253 timers, and the 8259 interrupt controller as described in the hardware chapter. The DAS 1128 is initialized so that probe "0" is the first to be sampled when the service routine series is started. The final tasks accomplished by the power-up routine are the initialization of the scratch pad storage and the initiation of the service routine series (program loop).

Error Routine. The operating system must ensure that the executing service routine is receiving data from its associated probe. Each service routine accomplishes this by checking the sensor identification information. If the service routine and sensor are mismatched, the operating system must select the proper sensor and return to the program loop. This is accomplished by the error routine.

Permanent Data Storage Routine. The operating system keeps track of the running time, and schedules the storage routine every ten seconds. The storage routine transfers a timing preamble and the data computed by the service routines from the scratch pad storage to the permanent storage device. The data is converted to the ASCII code required by the Hazeltine video terminal, and is transmitted over the RS232 interface. After all of the data is transferred, two control characters are transmitted which directs the Hazeltine system to write the data from the video screen onto the digital cassette tape. Finally, the scratch pad storage area is reinitialized. The data storage routine is the only device-dependent program in the operating system and is, therefore, the only module that will require modification when the new memory system is interfaced.

All of the service routines and system support software are stored on a single 2708 Erasable and Electrically Programmable Read Only Memory chip. This allows program modification by simply erasing and reprogramming the memory.

IV Post-Flight Data Conversion

The post-flight conversions are accomplished by a land-based computer after the flight. The data from the memory device is transferred to the computer where a program completes the data conversion and displays the desired information in graphic form. The form of this display (list, tabular, plot versus time, etc.) can be varied by changing the program.

This investigation utilized AFIT's computer system as the land-based computer. The data from the digital cassette tape is transferred to the computer (Appendix C) for the "post-flight" conversions. A FORTRAN program (Appendix D) reads in this data, converts it back to basic form, completes the calculation of the desired information, and displays this information in graphic form. Three environmental and five physiological parameters are graphed versus time by this conversion program. These are absolute pressure, cabin altitude, Z-G's, heart rate, breathing rate, minute ventilation volume, and inspired and expired oxygen volumes.

The following sections describe these calculations and include a representative graph of the parameter.

Absolute Pressure

The absolute pressure data is a number from 0 to 250 which is directly proportional to a pressure range of 0 to 760 mm Hg. The data is converted to actual cabin absolute pressure using the following formula:

$$\text{abs pressure (mm Hg)} = \frac{\text{(data)}}{250} \times 760 \qquad (1)$$

The cabin absolute pressure is plotted versus time, as in Figure 7.

Cabin Altitude

The absolute pressure data is also used to compute cabin altitude. There is not a simple mathematical relationship between data in mm Hg and altitude in feet (Ref 14:587); therefore, an approximate relationship was derived using linear regression techniques. A logarithmic curve of the form

$$y = a + b \ln(x) \qquad (2)$$

was found to provide the best fit. For the altitude range of 0 to 25,000 feet, the following equation was used:

$$\text{altitude (feet)} = 170{,}156 - 25{,}685 \ln(\text{abs pressure (mm Hg)}) \qquad (3)$$

This equation is accurate to within 275 feet for the 0 to 25,000 foot range. A sample cabin altitude versus time plot is shown in Figure 8.

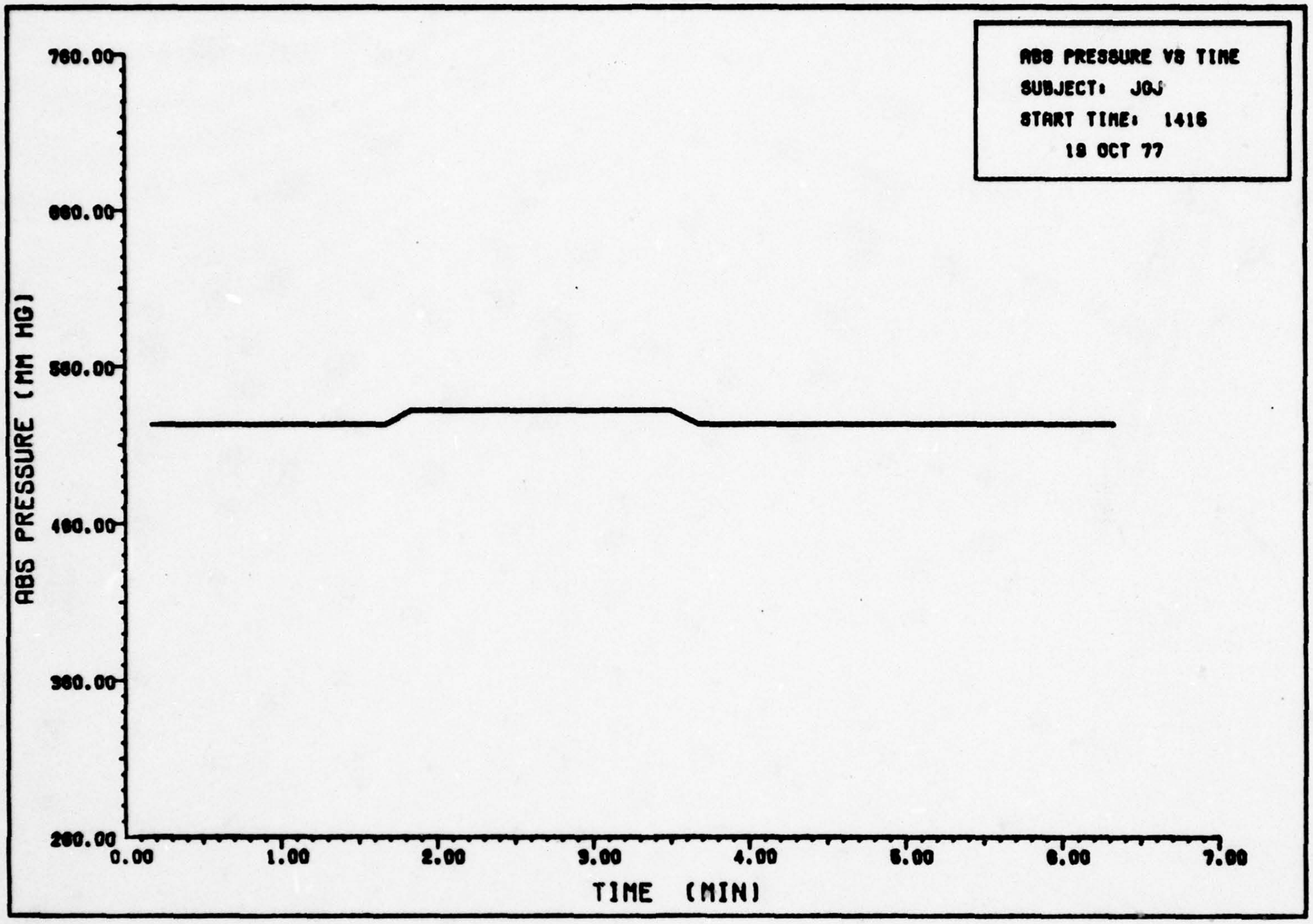


Fig. 7. Cabin Absolute Pressure vs Time

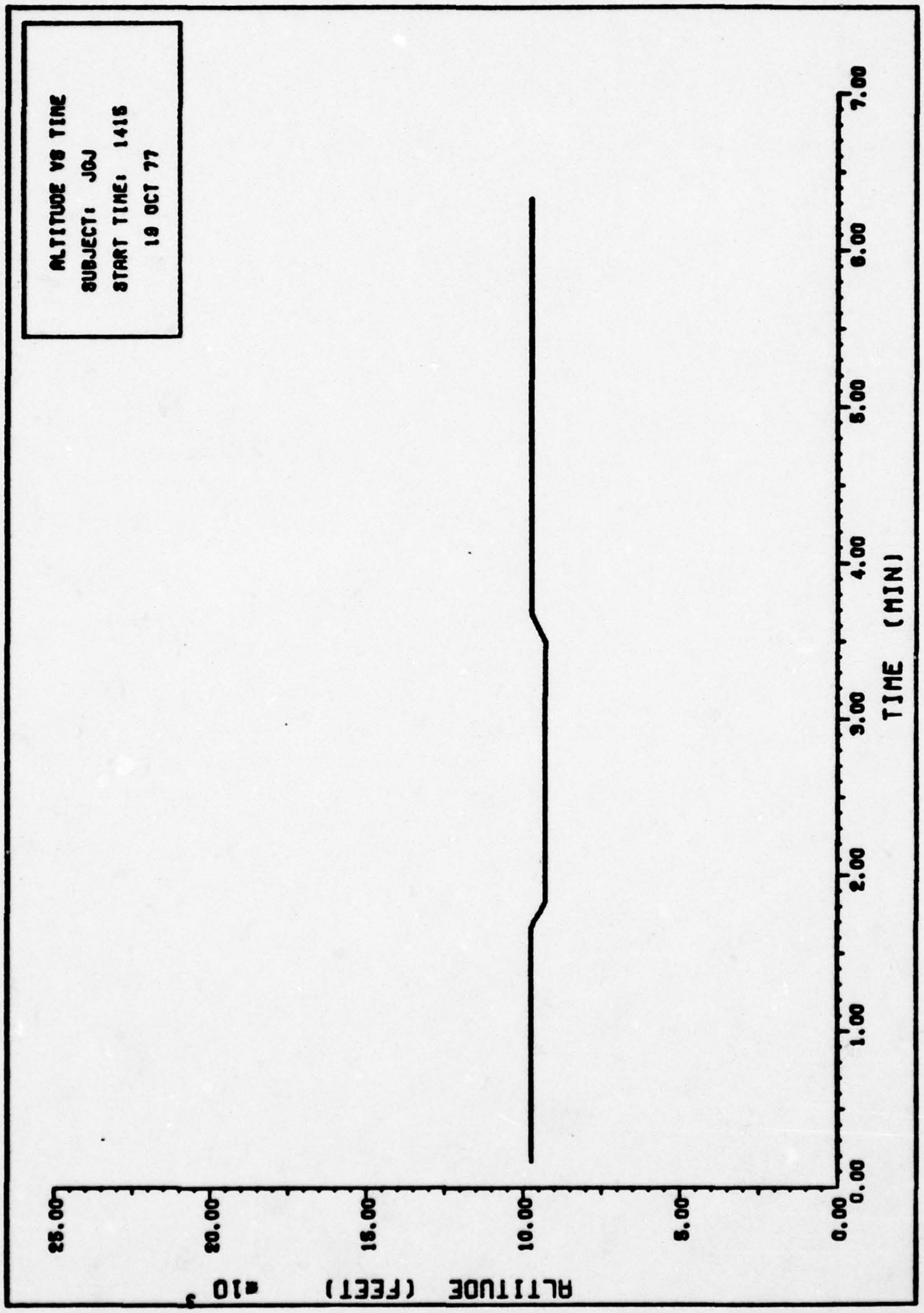


Fig. 8. Cabin Altitude vs Time

Z-G's

The acceleration data is a number from 0 to 250 which is directly proportional to a range of -3 to +12 G's. The data is converted to actual G's using the following formula:

$$\text{acceleration (G's)} = \left\{ \frac{\text{(data)}}{250} \times 15 \right\} - 3 \tag{4}$$

(The -3 term is necessary to correct for the 3 G offset of the acceleration circuitry.) Minimum and maximum Z-G's (for each ten second interval) are plotted on the same graph, as in Figure 9. (The graphs for X and Y-G's are similar and are not included.)

Heart Rate

The heart rate data is an average number of 4.44 msec (1/225 Hz) counts between detected R-waves. This number is converted to the heart rate using the following equation:

$$\text{heart rate (beats/min)} = \frac{1}{(4.44\ \text{msec})(\text{count})} \times 60$$

$$= \frac{225\ \text{Hz}}{(\text{count})} \times 60 \tag{5}$$

An example of the heart rate plot is shown in Figure 10.

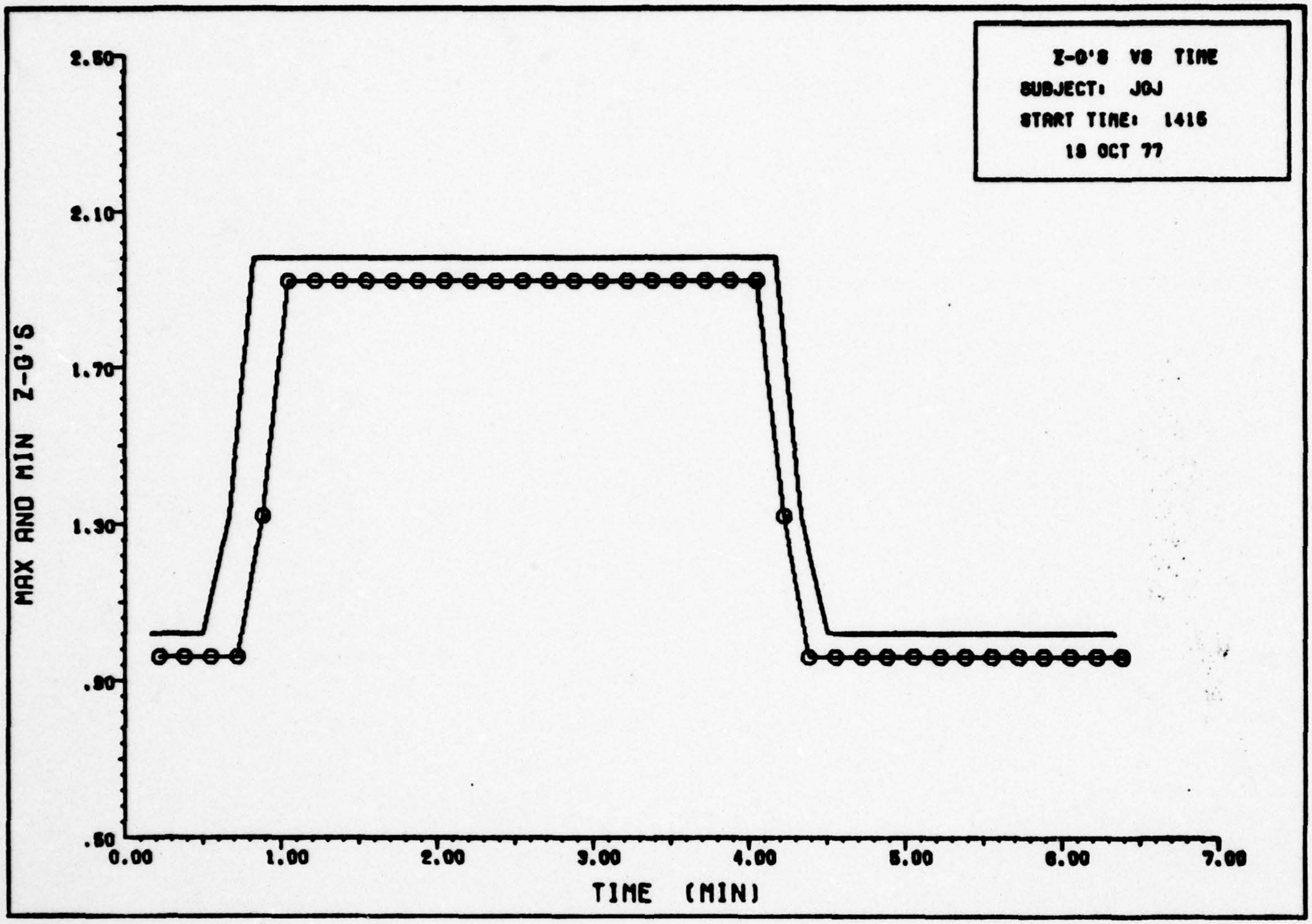


Fig. 9. Minimum and Maximum Z-G's vs Time

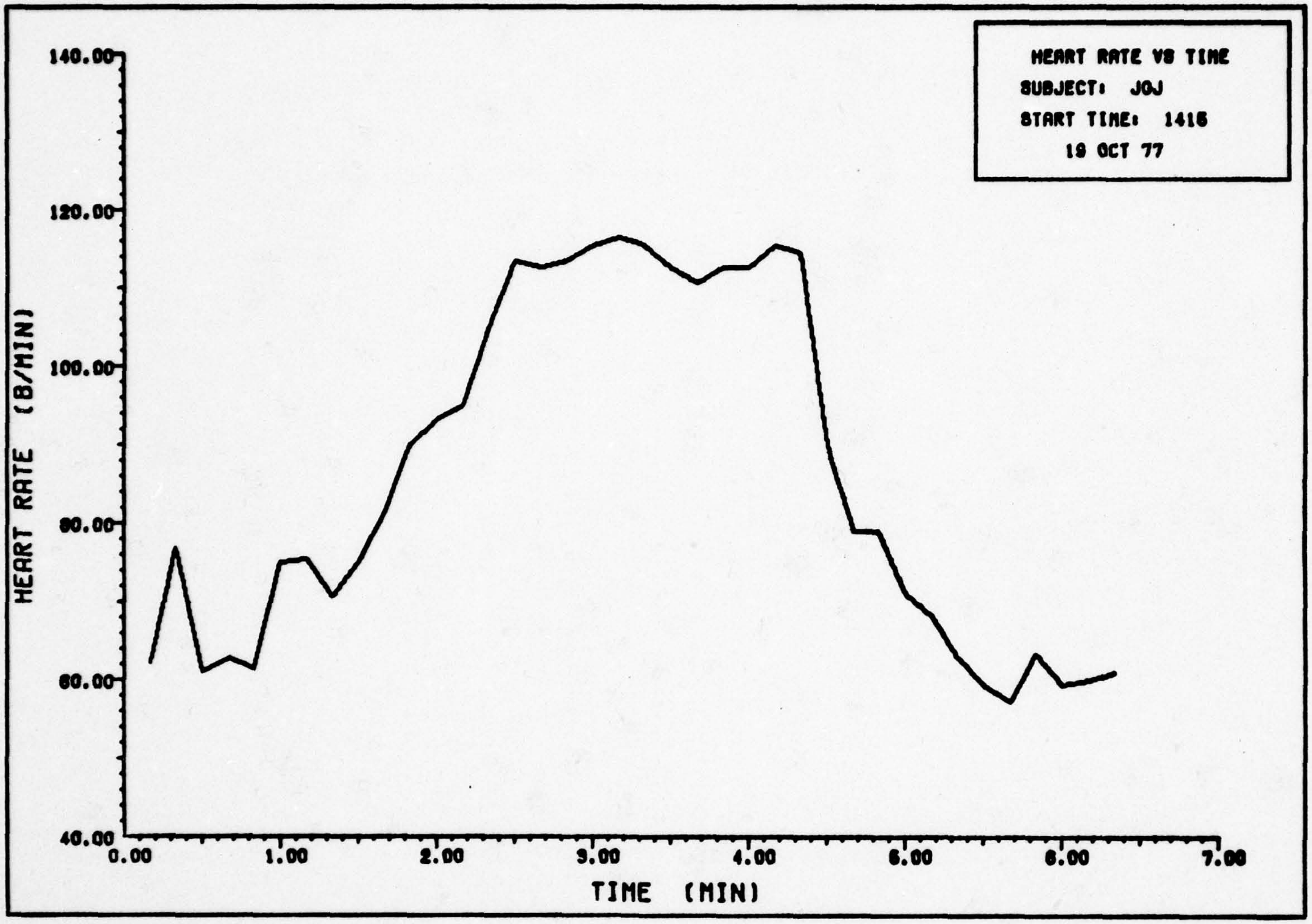


Fig. 10. Heart Rate vs Time

Breathing Rate

The breathing rate data is similar to the heart rate data, except that each count represents 50 msec. The breathing rate is calculated from the number of 50 msec counts in the following manner:

$$\text{breathing rate (breaths/min)} = \frac{1}{(50\ \text{msec})(\text{count})} \times 60$$

$$= \frac{20\ \text{Hz}}{(\text{count})} \times 60 \qquad (6)$$

The sample breathing rate graph is given in Figure 11.

Minute Ventilation Volume

The flow volume equations were derived for the original IFPDAS (Ref 9:5). The equation for flow volume rate (F) is

$$F = V_f\ (\text{volts}) \times (24.8\ \text{l/min/volt}) \times \sqrt{\frac{760\ \text{mm}}{\text{abs pressure}}} \qquad (7)$$

(The square root term corrects the flow volume rate for actual altitude.) The 50 msec volume flow rate reading (R) is a number from 0 to 250 which is directly proportional to a flow rate from 0 to 124 l/min. Each R could be used to compute an incremental flow volume expired during that 50 msec interval:

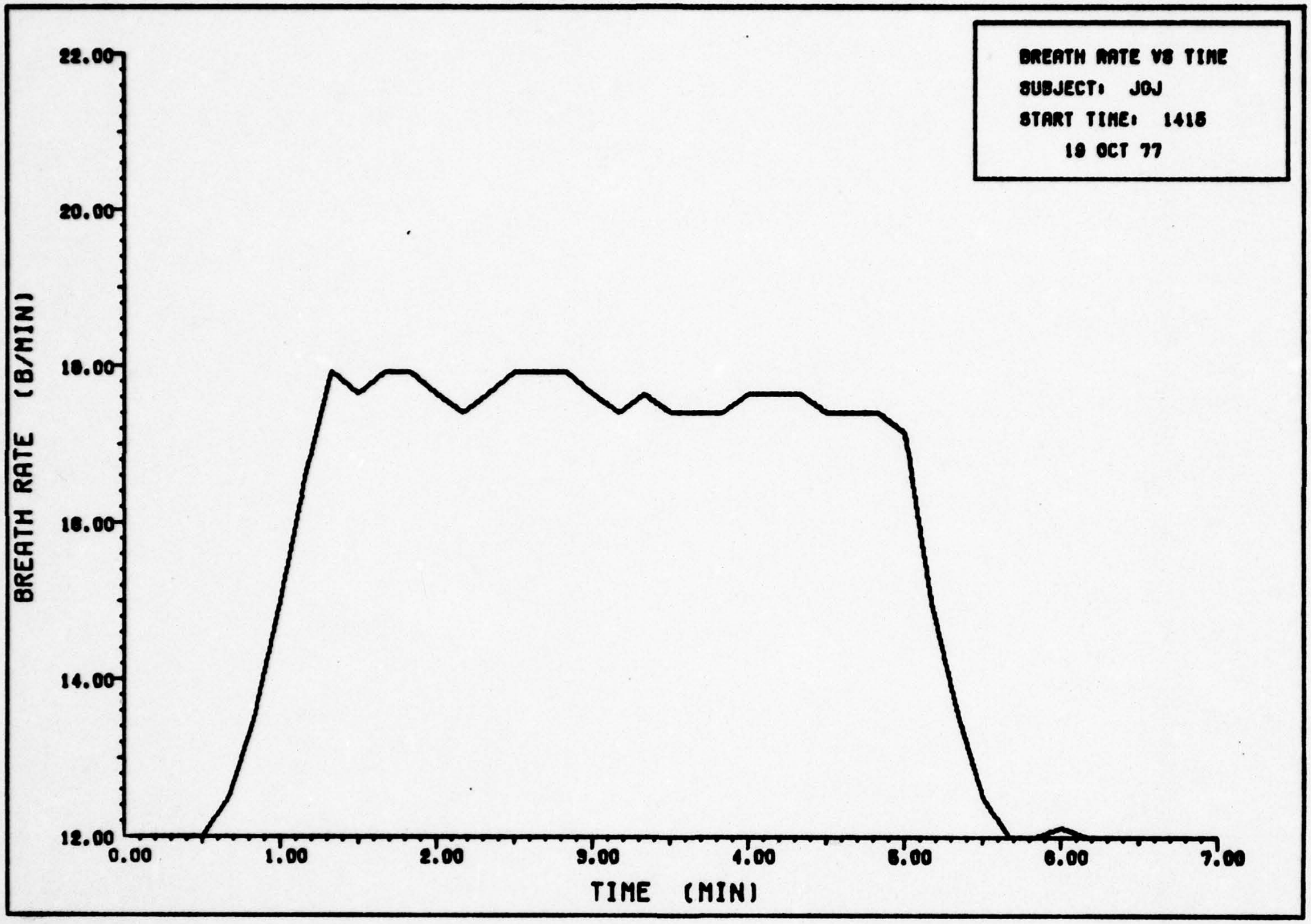


Fig. 11. Breathing Rate vs Time

$$\text{volume/50 msec (l)} = \frac{R}{250} \times (124\ \text{l/min}) \times \sqrt{\frac{760\ \text{mm Hg}}{\text{abs pressure}}} \times \frac{.05\ \text{sec}}{60\ \text{sec/min}} \qquad (8)$$

Summing each of the incremental volumes provides the flow volume for the ten second interval:

$$\text{volume/10 sec (l)} = \sum_{i=1}^{200} \left\{ \frac{R_i}{250} \times (124\ \text{l/min}) \times \sqrt{\frac{760\ \text{mm Hg}}{\text{abs pressure}}} \times \frac{.05\ \text{sec}}{60\ \text{sec/min}} \right\} \qquad (9)$$

or,

$$\text{volume/10 sec (l)} = \left\{ \frac{124\ \text{l/min}}{250} \times \sqrt{\frac{760\ \text{mm Hg}}{\text{abs pressure}}} \times \frac{.05\ \text{sec}}{60\ \text{sec/min}} \right\} \times \sum_{i=1}^{200} R_i \qquad (10)$$

The summation term is the value calculated and stored by the IFPDAS II prototype. Combining constants, Eq (10) reduces to

$$\text{volume/10 sec (l)} = .0114 \times \frac{\text{(summation)}}{\sqrt{\text{abs pressure}}} \qquad (11)$$

In order to obtain the desired minute ventilation volume, six 10-second volumes are summed. Minute ventilation volume is plotted versus time, as in Figure 12.

Oxygen Volumes

The inspired and expired oxygen partial pressure data is a sum of the 50 msec readings. This sum is divided by the number of readings, which gives an average oxygen partial pressure reading for the ten second period:

$$\text{avg reading} = \frac{\text{sum}}{200} \qquad (12)$$

This average reading is a number between 0 and 250 which is directly proportional to a pressure range of 0 to 760 mm Hg. This data is converted to the fractional amount of oxygen in the air as follows:

$$\text{fraction } O_2 = \left\{ \frac{\text{avg reading}}{250} \times 760 \right\} \Big/ \text{abs pressure} \qquad (13)$$

The quantity of oxygen in the inspired or expired air is then computed by

$$\text{quantity of } O_2 \text{ (l)} = (\text{fraction } O_2)(\text{minute vent vol (l)}) \qquad (14)$$

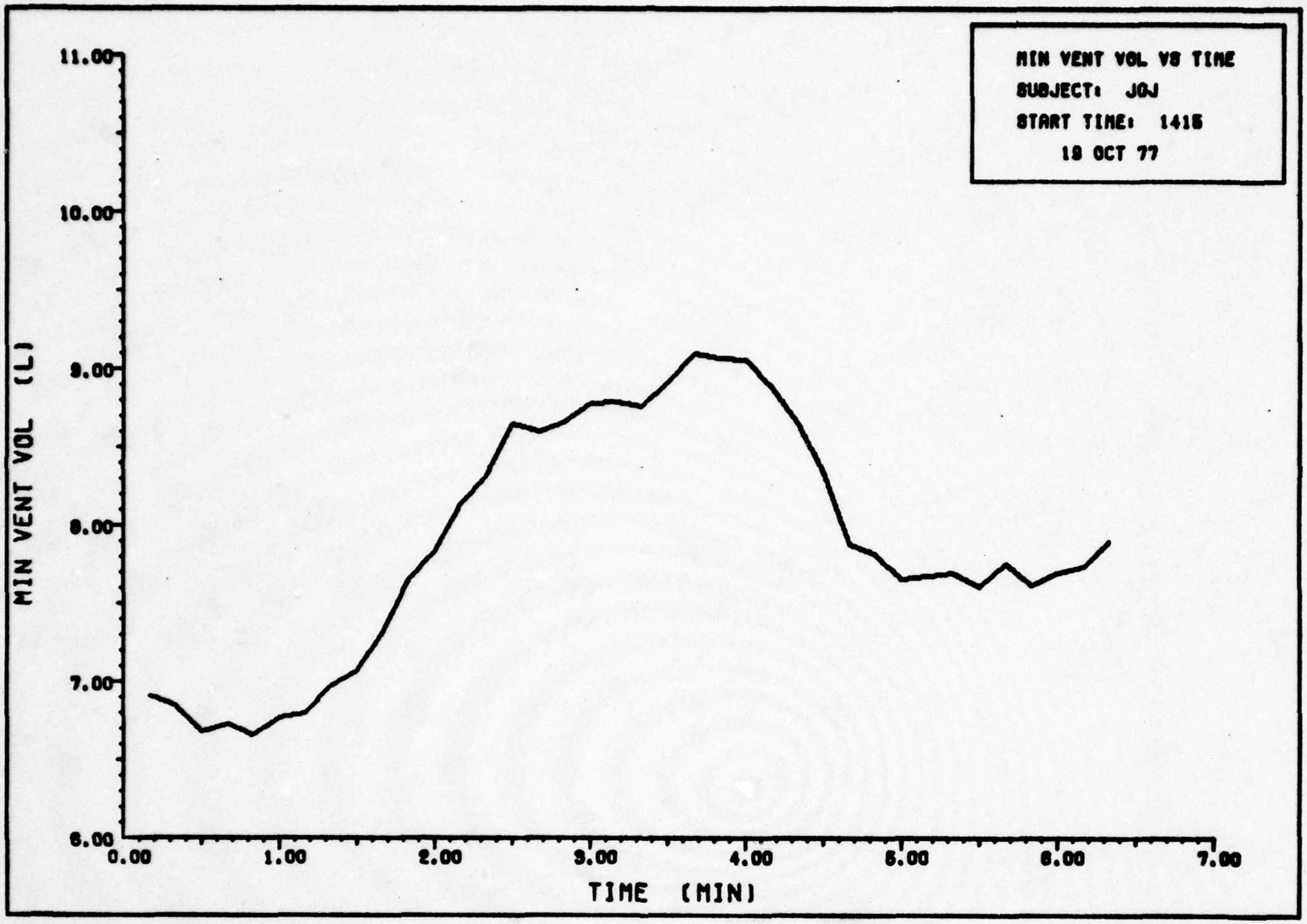


Fig. 12. Minute Ventilation Volume vs Time

Combining Eqs (12), (13), and (14)

$$\text{quantity of } O_2 \text{ (l)} = \frac{\text{sum X 760}}{\text{200 X 250 X (abs pressure)}} \text{ (min vent vol)} \qquad (15)$$

or,

$$\text{quantity of } O_2 \text{ (l)} = \frac{\text{(sum) X (min vent vol)}}{\text{65.79 X (abs pressure)}} \qquad (16)$$

The inspired oxygen quantity is graphed in Figure 13; and the expired oxygen quantity is graphed in Figure 14.

The graphs provide a means for easy correlation of the aircraft's pressure altitude and acceleration, and the pilot's heart rate, breathing rate, minute ventilation volume, and oxygen consumption.

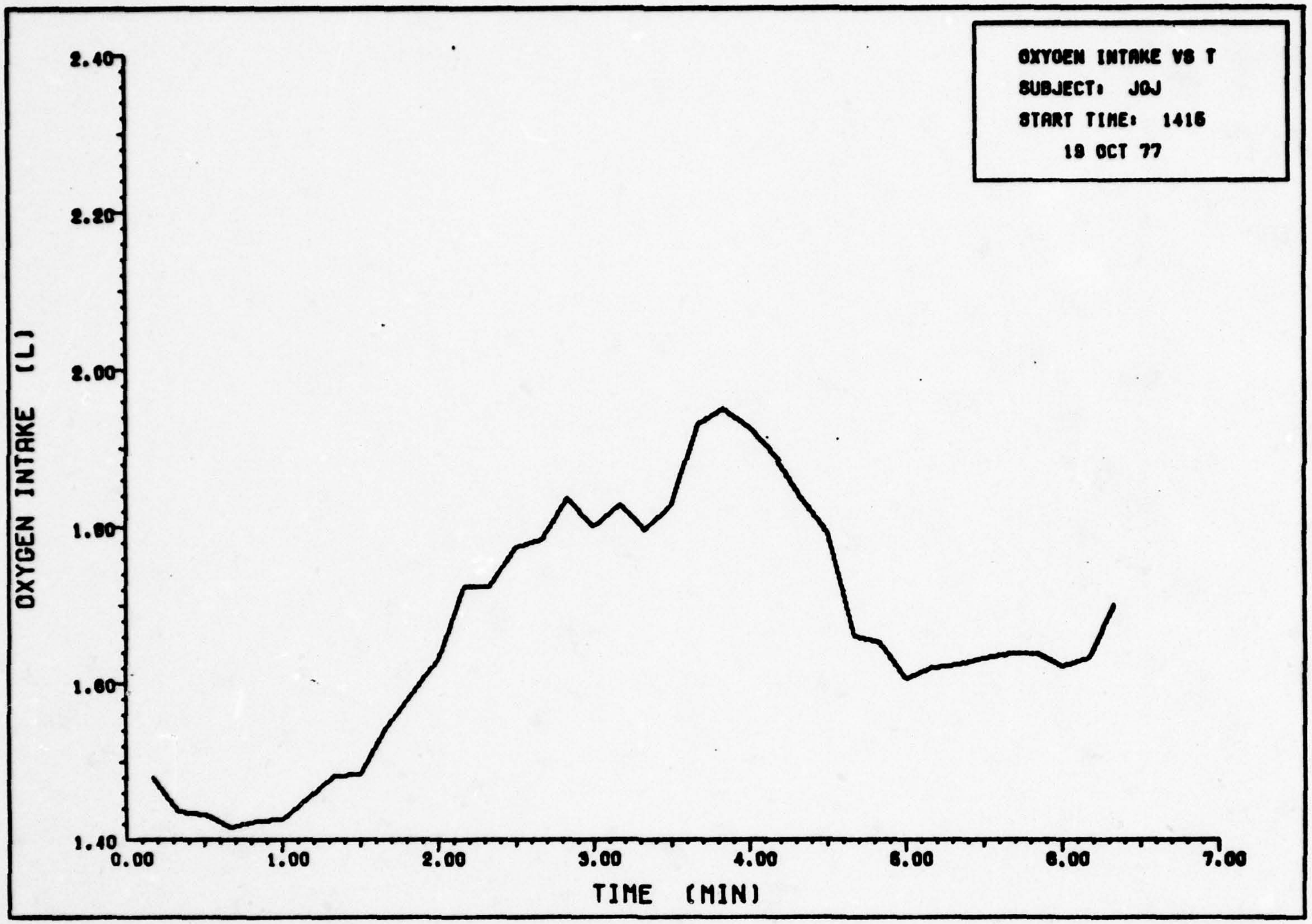


Fig. 13. Inspired Oxygen Quantity vs Time

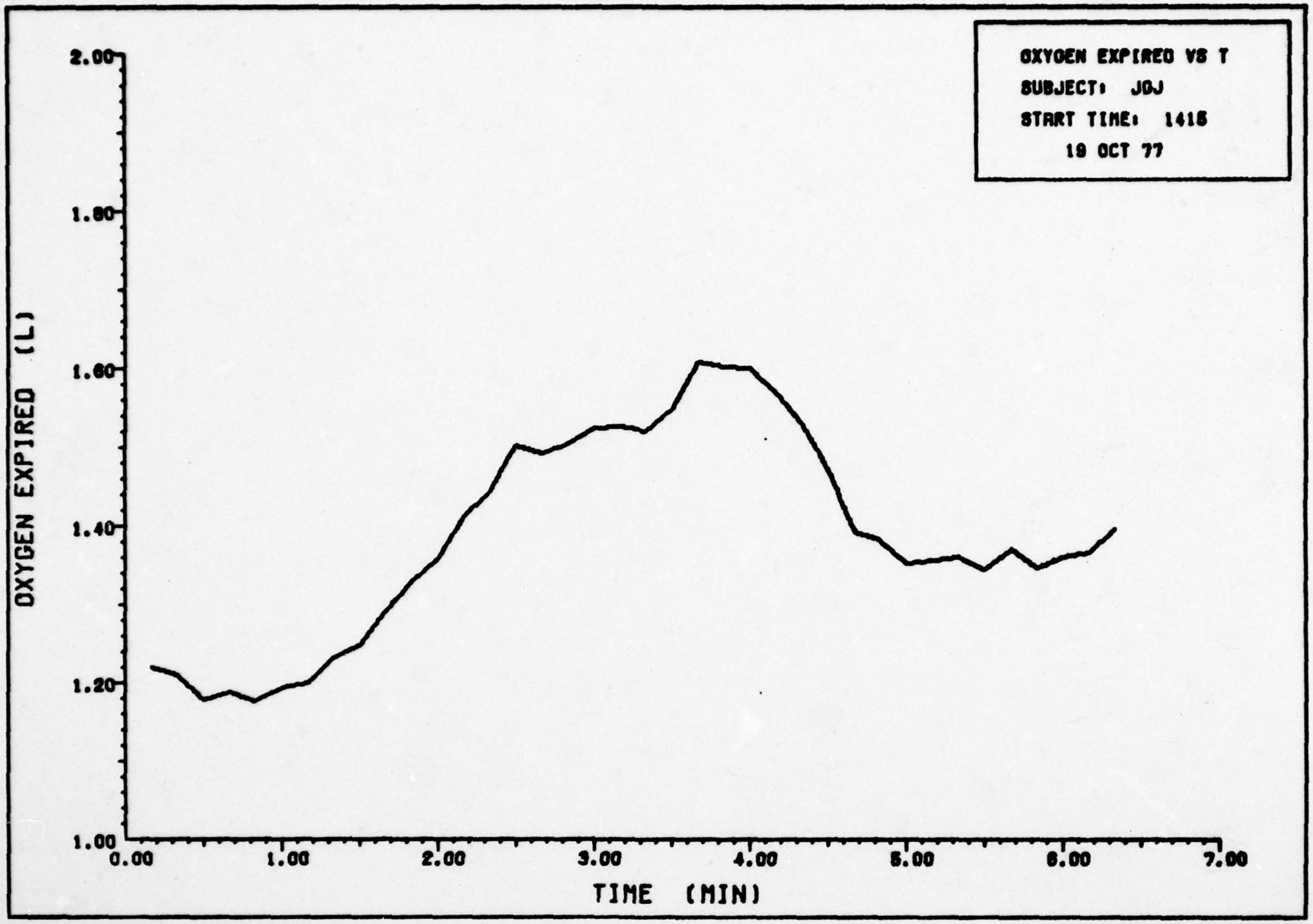


Fig. 14. Expired Oxygen Quantity vs Time

V System Tests

The IFPDAS II prototype was continually tested during development. Each program module, and each sensor interface, was checked individually both before and after system integration. After all of the modules and available interfaces were integrated, the IFPDAS II prototype was tested as a complete system. The following sections describe the developmental and final tests.

Developmental Tests

Each of the software modules was designed, coded, and tested on AFIT's main computer system (an 8080 simulator was used to verify the software). The tests consisted of inputing simulated data to check that the modules were manipulating the data properly. After these tests, the programs were transferred to the IFPDAS II prototype's memory.

Then the DAS 1128 was interfaced to the prototype and its conversion and input selection functions were checked. Initially, the $\overline{\text{STROBE}}$ and $\overline{\text{TRIG}}$ signals were supplied by digital switches which were used to manually step through the probe selections and start the A/D conversion. Then the operating system-supplied $\overline{\text{STROBE}}$ and $\overline{\text{TRIG}}$ signals (through the 8255) were used to perform the same functions. Known DC voltages were used as inputs to the signal multiplexer to check the accuracy of the conversion. For example: with the input voltage at 3.00 volts, the A/D converter output was 96H (150), where:

$$\frac{150}{250} \times 5.00 = 3.00 \qquad (17)$$

This testing ensured accurate conversion of the data by the DAS 1128 and proper manipulation of the data by the operating system.

After the operation of the DAS 1128 was verified, the data handling technique of each of the service routines was tested by applying a known DC voltage to the seven analog inputs. For the 3.00 volt example listed above, the routines that sum the readings over the 10 second interval (PO2IN, PO2OUT, and FLWRT) calculated a sum of 7530H (30,000), where,

$$\frac{30{,}000}{200} = 150 \qquad (18)$$

Also, the ABSPR routine stored a reading of 96H. To check the X, Y, and Z-G routines, two different DC voltages (3.00 and 4.00 volts) were used during the 10 second interval. Each of the routines stored 96H as the minimum reading and C8H (200) as the maximum reading. These tests showed that the prototype hardware was functioning as designed and that the operating system and its service routines were exercising the desired control and providing the desired data.

The error handling software was checked by forcing a sensor/service routine mismatch. The operating system identified this error and scheduled the ERROR routine which selected the proper sensor for the service routine. A test was also performed to determine how often the

ERROR routine was being executed. The system was run for four hours which provided over two million chances for a sensor/service routine mismatch. During the test period, the ERROR routine was never executed.

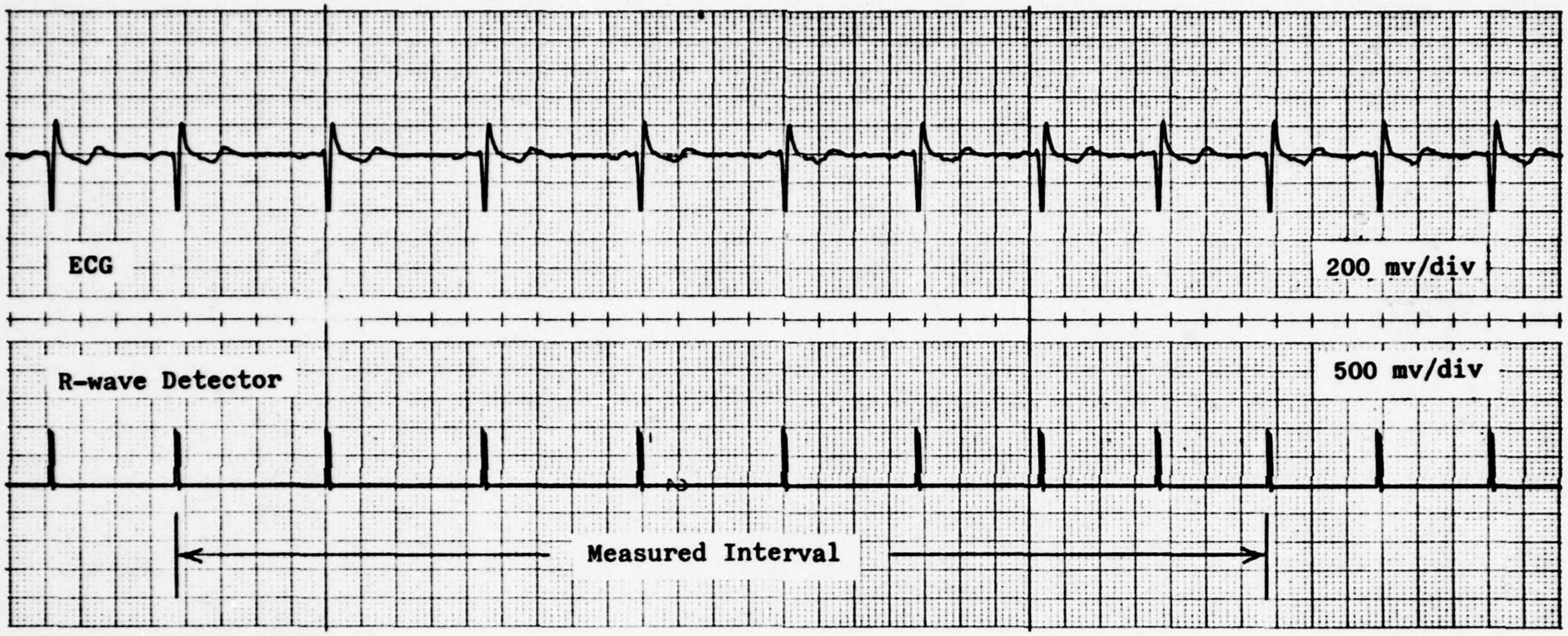
As each sensor interface circuit was developed, it was tested independently. It was then connected to the IFPDAS II prototype to test its integration with the system. Two sensors were not available (absolute pressure and expired flow rate); therefore, their hardware interface and software service routines could not be verified using actual sensor inputs. However, simulated inputs were used for the final tests.

The following sections describe the final tests of the completed IFPDAS II prototype and their results.

Final Tests

Heart Rate. The heart rate detector was checked in the following manner. ECG leads were attached to a subject and input to the detector. The output of the ECG amplifier was connected to channel 1 of a Gould Brush 440 strip chart recorder and the output of the R-wave detector was connected to channel 2. This allowed simultaneous recording of the ECG and R-wave detector outputs. Figure 15 shows a strip chart segment; and, a comparison of the IFPDAS II prototype calculated heart rate with a manual computation directly from the chart.

The IFPDAS II prototype uses eight R - R intervals to calculate a representative heart rate. The end of each averaging period was marked on the strip chart using the "mark-event" indicator; and the associated IFPDAS II calculation was recorded.



Manual Computation

Chart speed: 25 mm/sec
Distance for 8 beats: 155 mm
Time for 8 beats: 6.20 sec

Heart Rate: 77.4 b/min

IFPDAS Results

Average number of
4.44 msec counts: 175
Time for 1 beat: 0.777 sec

Heart Rate: 77.1 b/min

Fig. 15. Heart Rate Computation

The two calculations shown in Figure 15 are within .5% and are typical of the IFPDAS II prototype's accuracy (Appendix A). The differences between the two are due to variances in the chart speed and quantization of the count increment by the prototype.

Oxygen Partial Pressure. The oxygen partial pressure sensor amplifiers were calibrated to indicate 152 mm Hg (1.00 volt). The full scale range was then checked by exposing the OM11 sensor to a pure oxygen source. The amplifier output was 5 volts, indicating 760 mm Hg of oxygen. This 5 volt input (FAH) was properly interpreted and stored by the IFPDAS II operating system.

Acceleration. The Statham F-15-340 accelerometer was interfaced to the IFPDAS II prototype through its amplifier. The sensor axis was oriented to indicate acceleration in the vertical direction and the output of the amplifier was adjusted to 1.33 volts (+1 G). The accelerometer was then rotated 90^o to simulate a zero-G situation; and the output dropped to 1 volt (0 G's). Finally, the sensor was rotated an additional 90^o to simulate a negative G situation; and the output dropped to .67 volts (-1 G's). The IFPDAS II prototype recorded the minimum and maximum G's (within 0.1 G's) during each ten second interval of the test.

Absolute Pressure. As mentioned earlier, the absolute pressure probe was not available for use on the prototype. For this reason, a known DC voltage was used to simulate the absolute pressure input. As before, the operating system stored the proper value.

Flow Rate. Since the flow rate sensor was not available, a .20 Hz sine wave oscillator was used to supply the flow rate signal. The sine wave approximates the flow rate signal as shown in Figure 16. (The samples of the negative portion of the sine wave are "0" since the A/D converter input range is 0 - 5 volts.)

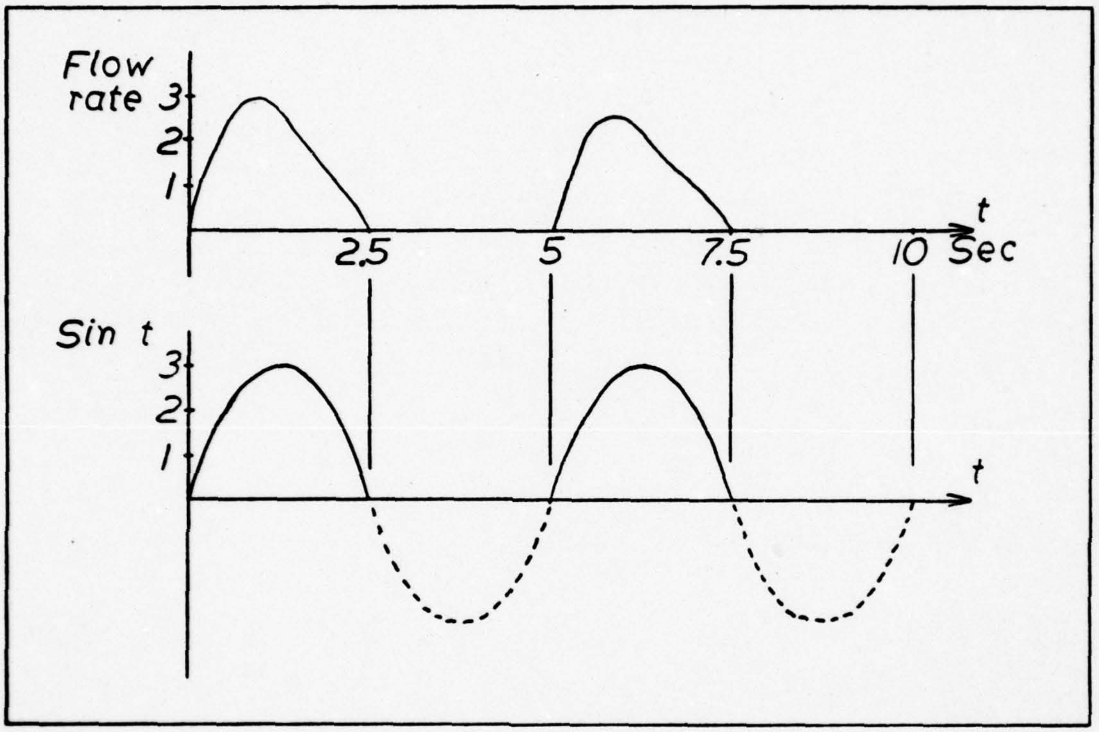


Fig. 16. Sine Wave Approximation of Flow Rate Signal

The amplitude of the sine wave was set to 3.0 volts and the frequency to .20 Hz using an oscilloscope. (The frequency was confirmed by the breathing rate - see the breathing rate test.) The sum of the 200 readings ranged from 2535H to 253EH, averaging 253AH (9530). The corresponding approximation of the integral is

$$(9530 \text{ counts}) \times .05 \text{ sec} = 476.50 \text{ count-sec} \qquad (19)$$

Since there are 50 counts/volt,

$$\frac{476.50 \text{ count-sec}}{50 \text{ counts/volt}} = 9.53 \text{ volt-sec} \qquad (20)$$

The exact integral yields:

$$2 \times \int_{0}^{2.5 \text{ sec}} 3 \sin(2\pi(.2)t)\, dt = \frac{12}{.4\pi}$$

$$= 9.55 \text{ volt-sec} \qquad (21)$$

The approximation of the integral is within .3% of the actual value. (This technique was also used to check the PO2IN and PO2OUT summing routines, with the same accuracy.)

Breathing Rate. The flow rate readings are used by the operating system to determine the breathing rate. During the flow rate test, the breathing rate routine recorded 64H (100) fifty msec counts per breath which is one breath every 5 seconds, or 12 breaths per minute.

The test results show that the IFPDAS II prototype monitors the sensor inputs accurately, and can extract and store the desired data. Appendix A summarizes these results and discusses the prototype's accuracy.

VI Suggestions and Recommendations

The BK10103 Magnetic Bubble Memory System and the TMS9916 controller should be interfaced to the IFPDAS II prototype and tested. This will allow evaluation of the IFPDAS II as a complete independent system.

The data handling technique for the bubble memory system will have to be developed. This includes a software routine to format the data and provide the commands necessary to store the data. In addition, a data retrieval method must be designed. This could include direct transfer from the bubble memory to a large land-based computer; or transfer of the data to a digital cassette tape. Transferring the data to cassette tape might be desirable in the field where access to a large computer is unlikely. The transfer of the entire memory contents would be accomplished by the IFPDAS II's 8080 microprocessor in a matter of minutes, thus, freeing the bubble memory for another mission. The data from the first mission is conveniently stored for later analysis.

The tradeoffs between a magnetic bubble memory system and digital cassette storage systems should be investigated further. This could include acquiring a cassette tape system and interfacing it to the IFPDAS. The performance of both systems could then be directly compared in terms of speed, ease of data transfer, power consumption, etc.

There are several methods to select the "important" data for reduction and storage, in addition to the one selected for the IFPDAS

II prototype. Two others are a breath-by-breath analysis and a "store-on-significant-change" technique. The breath-by-breath analysis would be similar to the method used in the prototype; except that the data would be stored every breath instead of every ten seconds. A digital differentiation technique could be employed to detect significant changes in selected parameters. The operating system would then analyze the change, then collect and store any required data. A thorough analysis of expected flight profiles must be made in order to determine which of the acquisition techniques, or combination of techniques, would be most desirable for a given profile. (This profile analysis may also effect the final choice of memory system.)

An absolute pressure sensor and a bi-directional flow sensor should be acquired and interfaced to the IFPDAS II prototype. With these sensors, and the present OM11 sensors, a more accurate analysis of the oxygen consumption can be made. A suggested method incorporating a single breath analysis is given in Appendix I.

The prototype should be reduced to the IFPDAS II size restrictions. This will include eliminating the unnecessary components on the SBC 80/20 board; and should include consideration of newly developed chips which combine the functions of several chips into one unit. (An example is Intel's new 8085 microprocessor and component family. The components on the SBC 80/20 board that are needed for the IFPDAS II system can be replaced by three 8085 system chips. These components use only 3 watts of power, can be placed on a 3.25" x 4.5" printed circuit board, and are 100% software compatible with the 8080 (Ref 6:109-113).) Once the configuration is reduced, the power consumption should be carefully considered. At this time, the complete IFPDAS II can be evaluated against the design criteria specified by SAM.

Other areas that should be investigated further include: the capability to record the pilot's voice and synchronize it with the data; a probe check routine which would give a visual indication if the probes were not properly interfaced; and a standard test device to bench-check the entire IFPDAS II operation.

Bibliography

1. Durand, D., Katona, P., and Lin, W. C. "Microprocessor Controlled Memory for Cardiopulmonary Monitoring of High Risk Infants," paper presented at the 28th ACEMB, Fairmont Hotel, New Orleans, La, 20 - 24 September 1975.

2. Moritz, W. E. and Murdock, D. B. "Microprocessor Based Patient Monitoring System," paper presented at the 28th ACEMB, Fairmont Hotel, New Orleans, La, 20 - 24 September 1975.

3. Carroll, D. R. and Demitry, L. F. "Microprocessor Implementation of an Automatic Blood Pressure Monitoring System," paper presented at the 28th ACEMB, Fairmont Hotel, New Orleans, La, 20 - 24 September 1975.

4. Weaver, Charles S., Guilleminault, L. F., and Harper, R. M. Research and Development of an Instrument to Monitor Cardiorespiratory Function in the Home. Menlo Park, California: Stanford Research Institute, May 1977.

5. Cushman, Robert H., ed. "EDN's Third Annual Microprocessor Directory," EDN, 21 (21): 44-89 (20 November 1976).

6. Sohn, D. W. and Volk, A. "Third Generation Microcomputer Set Packs It All Into 3 Chips," Electronics, 50 (10): 109-113 (12 May 1977).

7. MCS 85 Users Manual. Product catalog. Santa Clara, California: Intel Corporation (March 1977).

8. Osborne, Adam. An Introduction to Microcomputers. Berkeley, California: Adam Osborne and Associates, Inc., 1976.

9. Contract/PR F41609-73-C-00009. Aircrew Inflight Physiological Data Acquisition System. System description. Houston, Texas: SCI Systems, Inc., March 1974.

10. Powers, SMSgt Joseph. Impact Branch, AMRL. Wright-Patterson AFB, Ohio (personal interview) 14 June 1977.

11. Microvox: Software-Interface Digital Data Storage Systems. Product catalog. Waltham, Massachusetts: Micro Communications Corporation (April 1977).

12. Digital Cassette Tape Transport. Technical data. Salem, New Hampshire: MFE Corporation (December 1975).

13. Lee, David M. "Bubble Memory for Microprocessor Mass Storage". Article published in Magnetic Bubble Memories and System Interface Circuits. Dallas, Texas: Texas Instruments, Inc., 1977.

14. Guyton, Arthur C. Textbook of Medical Physiology (Fifth Edition). Philadelphia: W. B. Saunders Co., 1976.

15. SBC 80P20 Users Guide. Santa Clara, California: Intel Corporation (1977).

16. SBC 80/20 Single Board Computer Hardware Reference Manual. Santa Clara, California: Intel Corporation (1976).

17. DAS 1128: Low Cost, High Speed Data Acquisition Module. Product information. Norwood, Massachusetts: Analog Devices (1976).

18. Rost, Gerald A. "Recent Improvements in Hypoxia Warning Systems," Aerospace Medicine, 41 (8): 865-868 (August 1970).

19. -----. Beckman Instruments, Inc. (telephone interview) 6 October 1977.

20. Semiconductor Data Library, Volume 6. Linear integrated circuits. Motorola, Inc., 1975.

21. Fenn, Wallace O. and Rahn, H., eds. Handbook of Physiology, Volume 1, Section 3, "Respiration". Washington, D. C.: American Physiological Society, 1964.

22. MIL-R-25410F. Regulators, Oxygen, Diluter-Demand, Automatic Pressure Breathing. Washington: Department of Defense, 15 January 1968.

23. Baumgardner, F. Wesley, PhD., Research Physiologist. Crew Technology Division, USAF SAM. Brooks AFB, Texas (telephone interview) 14 October 1977.

24. AN-714. A Personalized Heart Rate Monitor With Digital Readout. Motorola Applications Note. Motorola, Inc., 1975.

25. Welkowitz, Walter and Deutsch, S. Biomedical Instrumentation Theory and Design. New York: Academic Press, 1976.

26. Connelly, Joseph A. Analog Integrated Circuits. New York: John Wiley and Sons, 1975.

27. Slonim, N. Balfour and Chapin, John L. Respiratory Physiology. St Louis: The C. V. Mosby Company, 1967.

28. Garrett, Patrick. "Optimize Transducer/Computer Interfaces," *Electronics Design*, *11*: 178-181 (24 May 1977).

29. Celler, B. G. and Bason, P. T. "Hardware Detection of P-waves in the Electrocardiogram," *Medical and Biological Engineering*: 501-507 (September 1976).

30. James, Gordon W., Paul, M. H., and Wessel, H. U. "Precision Digital Heart Rate Meter," *Journal of Applied Physiology*, *32* (5): 718-723 (May 1972).

31. *The TTL Data Book for Design Engineers* (Second Edition). Dallas, Texas: Texas Instruments, Inc. (1976).

Appendix A

System Characteristics

Table I

Parameter Ranges

Parameter	Desired Range	IFPDAS II Range
Heart Rate	50 to 180 ± 2 b/min	53 ± .1 to 225 ± 2.2 b/min
Breathing Rate	10 to 30 ± 2 b/min	4.7 ± .1 to 50 ± 1 b/min
Flow Rate	N/S	0 to 124 ± .25 l/min *
Flow Rate Integral	N/S	0 to 26.2 volt-sec (± .3%)
Minute Ventilation Volume	5 to 35 ± 2 l/min	0 to 100 ± 2 l/min **
Oxygen Partial Pressure	N/S	0 to 760 ± 2 mm Hg
Absolute Pressure	N/S	0 to 760 ± 2 mm Hg *
G's	-3 to +12 ± .25 G's	-3 to +12 ± .1 G

N/S = Not Specified

*Assuming accurate probe input and ± ½ bit accuracy (± 10 mv)

**At sea level, assuming an accurate probe input

Table II

Prototype Power Supplies

SBC 80/20	+ 5 VDC ± 5%	@ 3.5 Amps
	- 5 VDC ± 5%	@ .180 Amps
	+ 12 VDC ± 5%	@ .467 Amps
	- 12 VDC ± 5%	@ .123 Amps
DAS 1128 and Sensor Interfaces:	+ 15 VDC ± 3%	@ .050 Amps
	- 15 VDC ± 3%	@ .100 Amps
	+ 5 VDC ± 3%	@ .500 Amps

Table III

Memory Map

Hexadecimal Address	Use
0000 - 03DF	Operating System
03E0 - 03FF	Interrupt Jump Table
0400 - 0FFF	Unused PROM
1000 - 37FF	Not Used
3800 - 3BFF	Unused RAM
3C00 - 3C2F	CPU Scratch Pad Area
3C30 - 3F4F	Unused RAM
3F50 - 3F80	CPU Stack Area
3F81 - 3FFF	Unused RAM

Table IV

Pin Connections for DAS 1128 Interface

SBC 80/20 Board		Component Board		
Signal	J1 Pin Connection	J4 Pin Connection	Signal	
Port 2 - Bit 7	2	52	8 Out	Mux Addr
Bit 6	4	54	4 Out	
Bit 5	6	56	2 Out	
Bit 4	8	58	1 Out	
Bit 3	10	60	B9	
Bit 2	12	62	B10	
Bit 1	14	64	B11	
Bit 0	16	66	B12 (LSB)	
IBFA	18	68	N/C	
$\overline{\text{STROBE}}$	20	70	$\overline{\text{STROBE}}$	
IBFB	22	72	N/C	
$\overline{\text{TRIG}}$	24	74	$\overline{\text{TRIG}}$	
$\overline{\text{STBA}}$	26	76	$\overline{\text{EOC}}$	
N/C	28	78	N/C	
N/C	30	80	N/C	
$\overline{\text{STBB}}$	32	82	$\overline{\text{EOC}}$	
Port 1 - Bit 7	34	84	B1 (MSB)	
Bit 6	36	86	B2	
Bit 5	38	88	B3	
Bit 4	40	90	B4	
Bit 3	42	92	B5	
Bit 2	44	94	B6	
Bit 1	46	96	B7	
Bit 0	48	98	B8	
N/C	50	100	N/C	

Odd numbered pins are GND

Table V

Pin Connections for Heart Rate Detector Interface

SBC 80/20 Board		Component Board	
Signal	J2 Pin Connection	J4 Pin Connection	Signal
Port 2 - Bit 7	2	2	N/C
Bit 6	4	4	N/C
Bit 5	6	6	N/C
Bit 4	8	8	N/C
Bit 3	10	10	N/C
Bit 2	12	12	N/C
Bit 1	14	14	N/C
Bit 0	16	16	N/C
IBFA	18	18	N/C
N/C	20	20	N/C
IBFB	22	22	N/C
N/C	24	24	N/C
$\overline{\mathrm{STBA}}$	26	26	$\overline{\mathrm{STB}}$
N/C	28	28	N/C
N/C	30	30	N/C
$\overline{\mathrm{STBB}}$	32	32	N/C
Port 1 - Bit 7	34	34	B1 (MSB)
Bit 6	36	36	B2
Bit 5	38	38	B3
Bit 4	40	40	B4
Bit 3	42	42	B5
Bit 2	44	44	B6
Bit 1	46	46	B7
Bit 0	48	48	B8 (LSB)
225 Hz Clock	50	50	Clock

Odd numbered pins are GND

Table VI

Hazeltine Interface Connections

Signal	J3 Pin Connection	Hazeltine Connection	Signal
Protective GND	2	1	Protective GND
Transmit Data	4	2	Transmit Data
Receive Data	6	3	Receive Data
Request to Send	8		N/C
Clear to Send	10		N/C
Signal GND	14	7	Signal GND
		5	Clear to Send
+12 volts	22	6	Data Set Ready
		8	Data Carrier Detect

Unused pins are not shown

Table VII

Other Characteristics

Maximum analog inputs : 16
Analog input impedence : $>10^{10}$ ohms
Analog input voltage : 0 - 5.12 volts
Sampling rate : 20 Hz
Machine cycle time : 465 nsec
Timer clock period : 930 nsec
Heart rate clock : 225 Hz
Transfer (baud) rate : 1200 baud

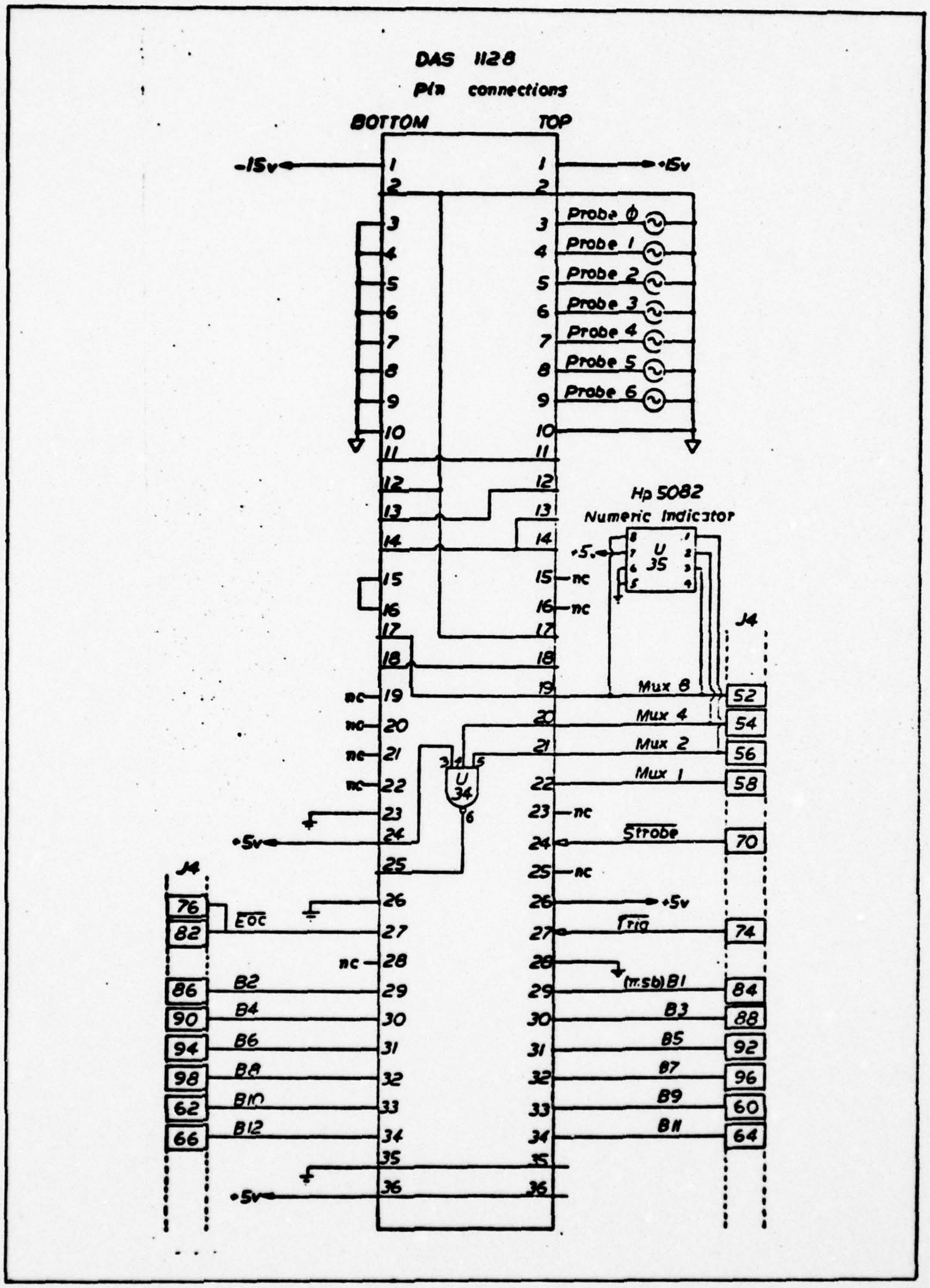


Fig. 17. DAS 1128 Interconnections and Interface

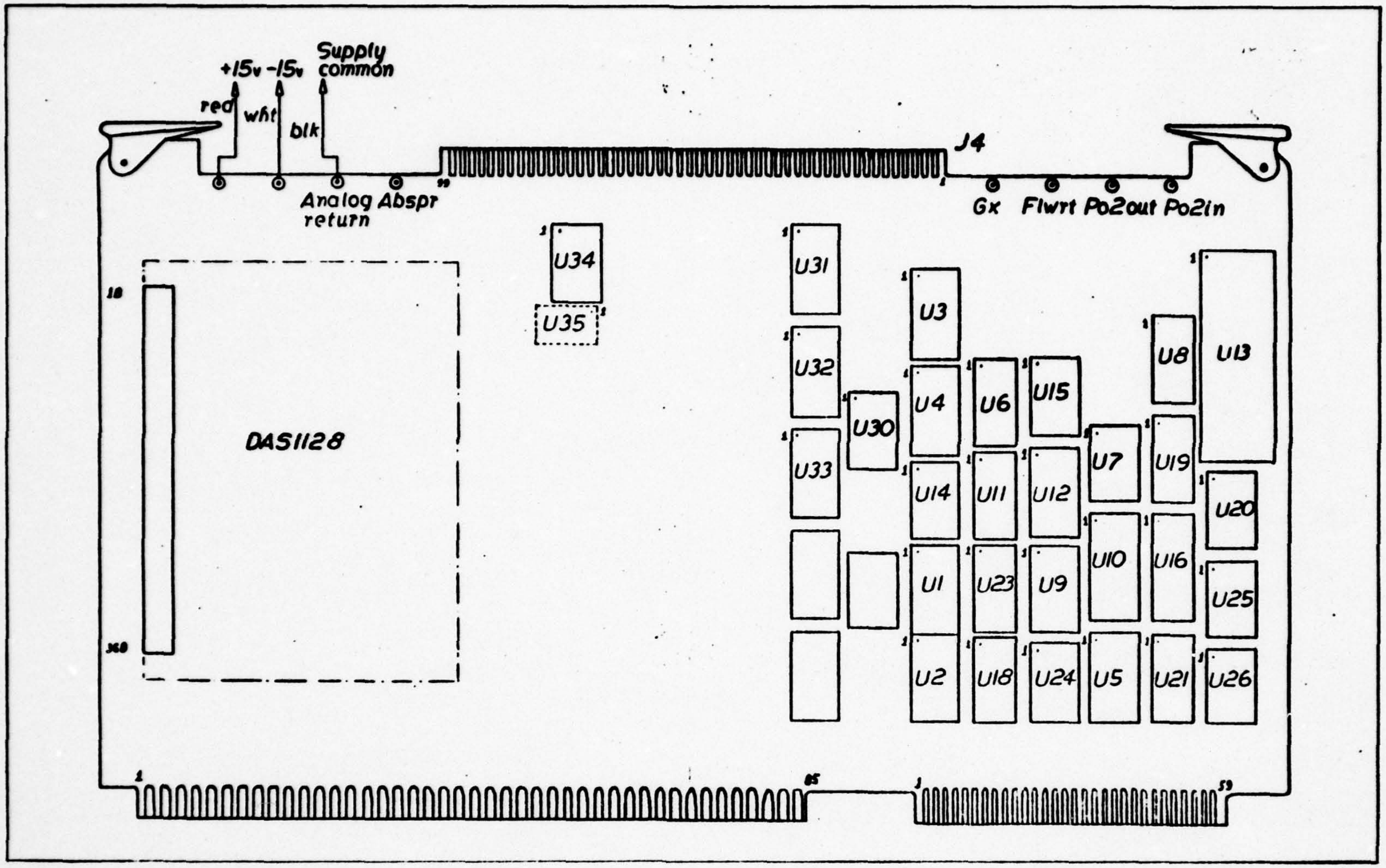


Fig. 18. Component Board Layout

Appendix B

Operating System

The operating system controls the operation of the IFPDAS II prototype hardware. Execution of the operating system by the CPU results in initialization of the system and acquisition, reduction, and storage of the data. The operating system selects the proper input sensor, starts the A/D conversion, and inputs the data for reduction and permanent storage. In addition, it ensures that the proper service routine receives the data.

A flow chart of the operating system starts on this page; a listing of the program follows the flow chart.

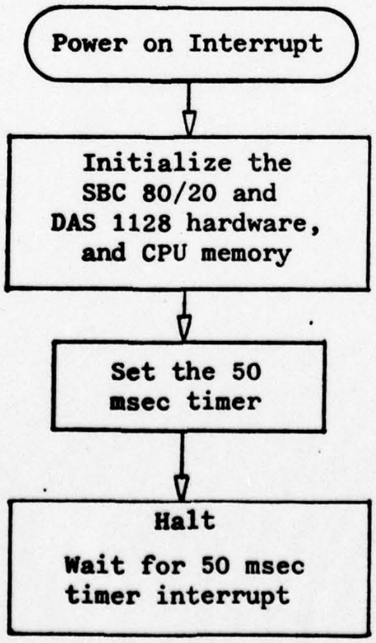


Fig. 19. Operating System Flow Chart (Sheet 1 of 9)

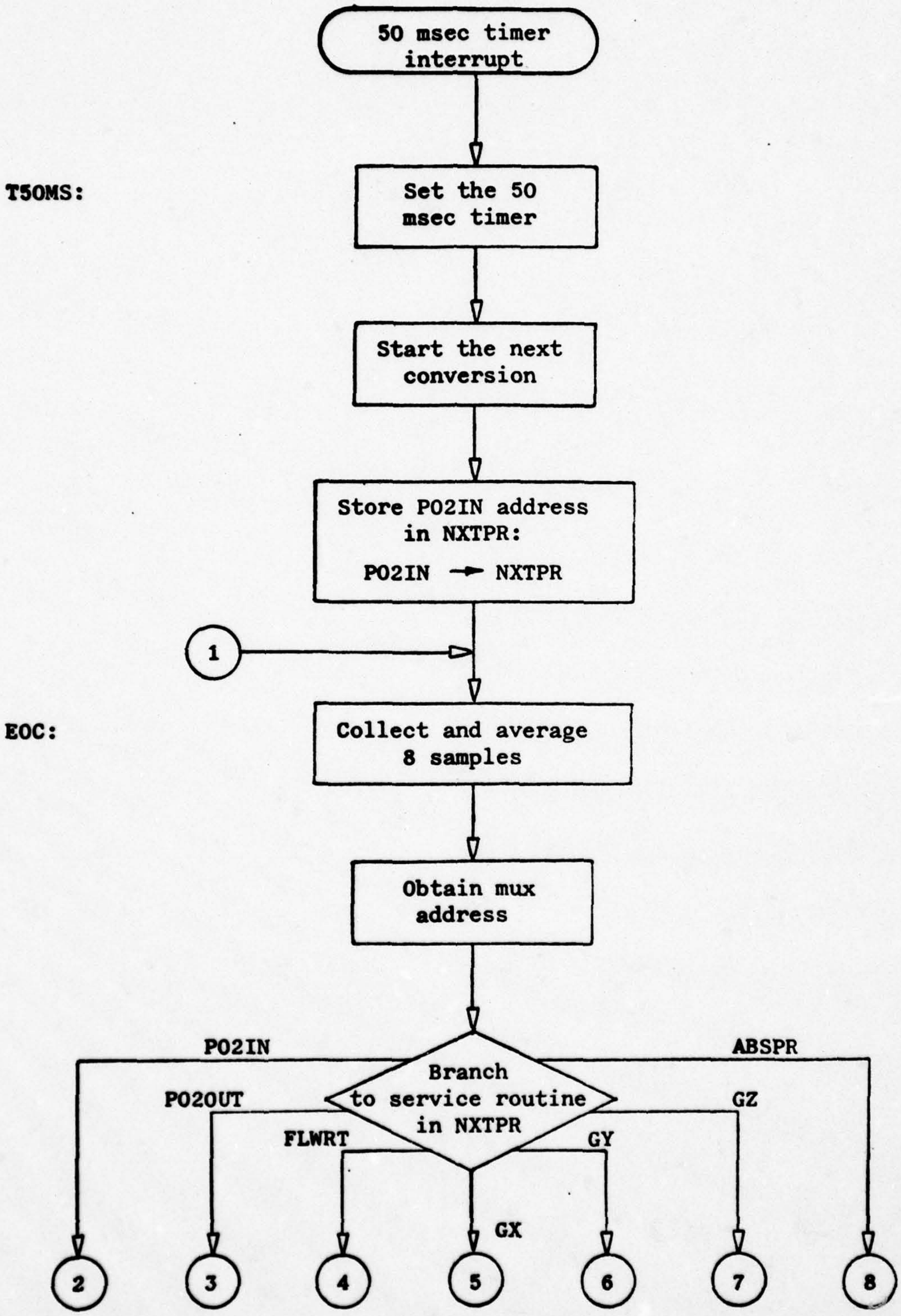
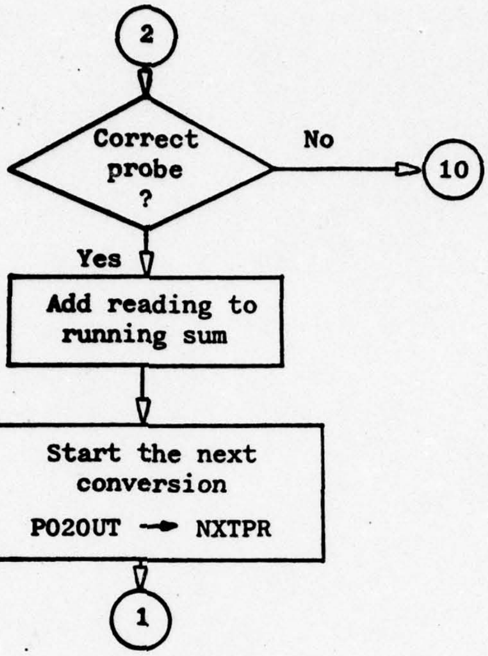


Fig. 19. Operating System Flow Chart (Sheet 2 of 9)

P02IN:



P02OUT:

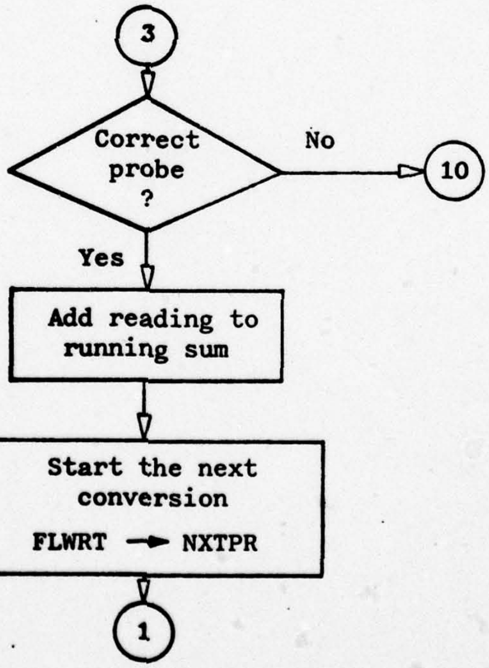


Fig. 19. Operating System Flow Chart (Sheet 3 of 9)

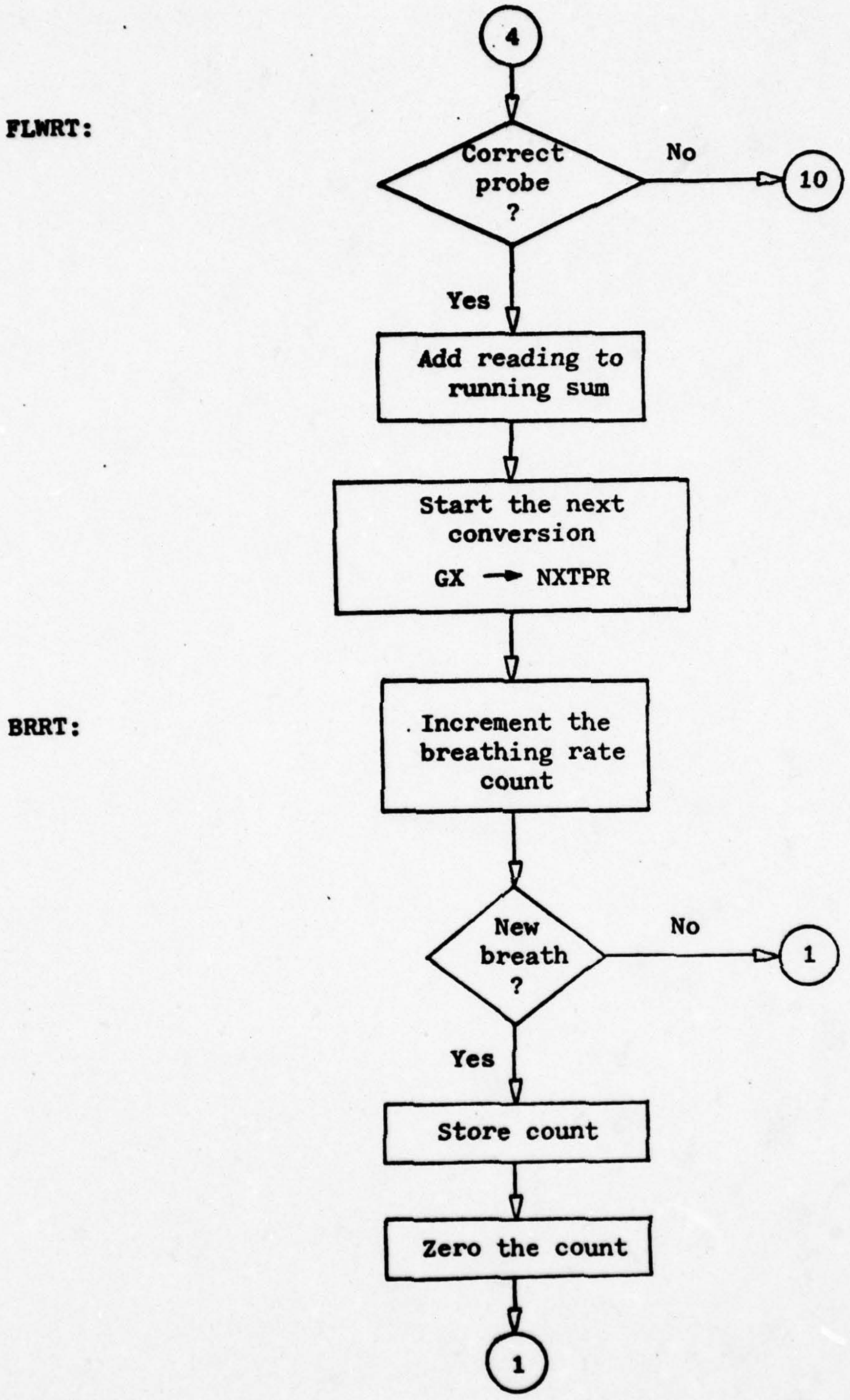


Fig. 19. Operating System Flow Chart (Sheet 4 of 9)

GX:

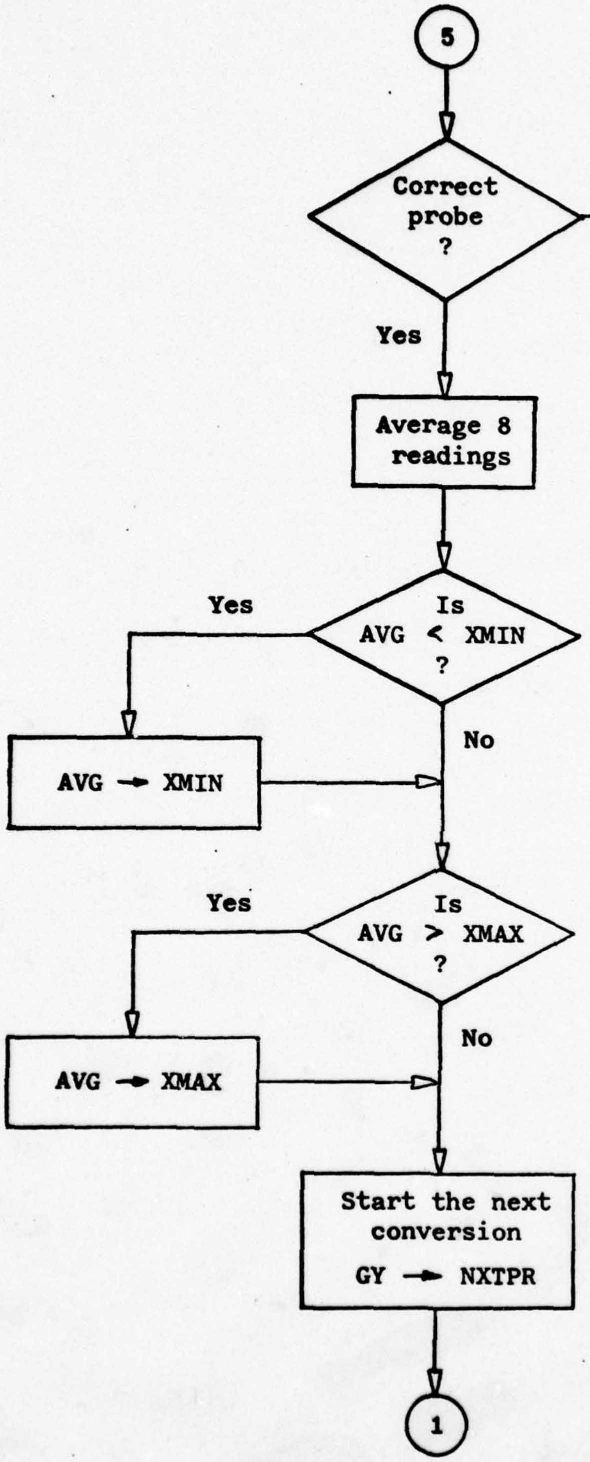


Fig. 19. Operating System Flow Chart (Sheet 5 of 9)

GY:

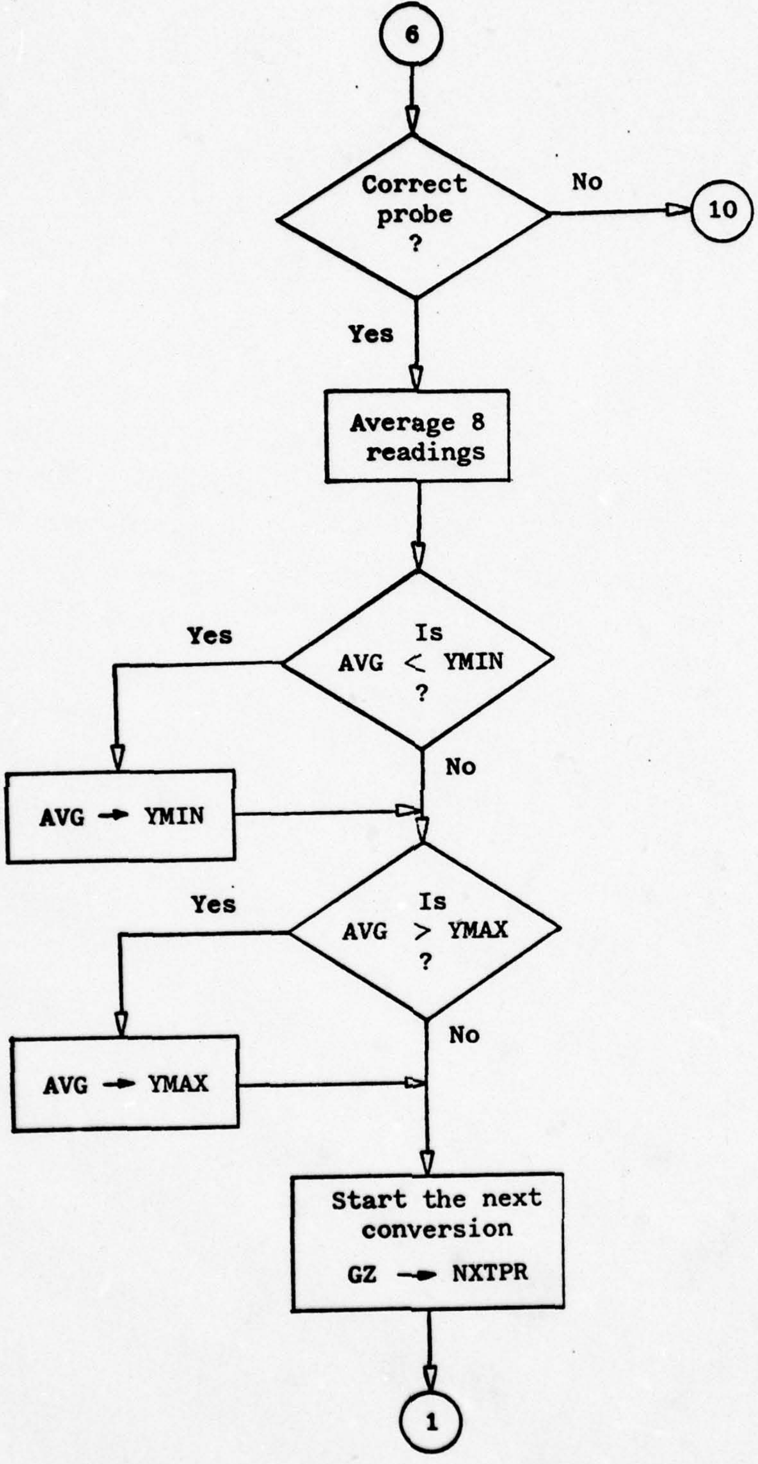


Fig. 19. Operating System Flow Chart (Sheet 6 of 9)

GZ:

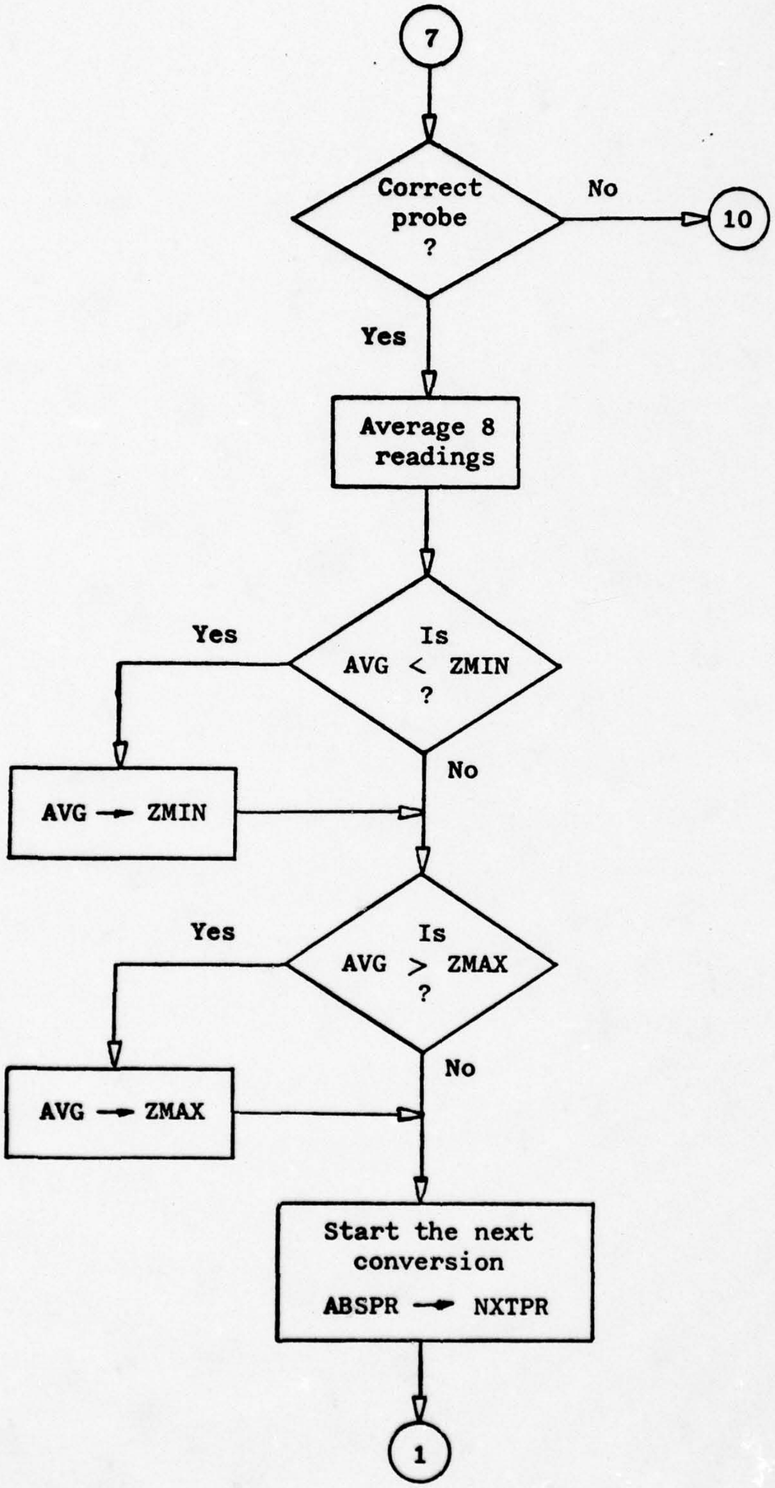


Fig. 19. Operating System Flow Chart (Sheet 7 of 9)

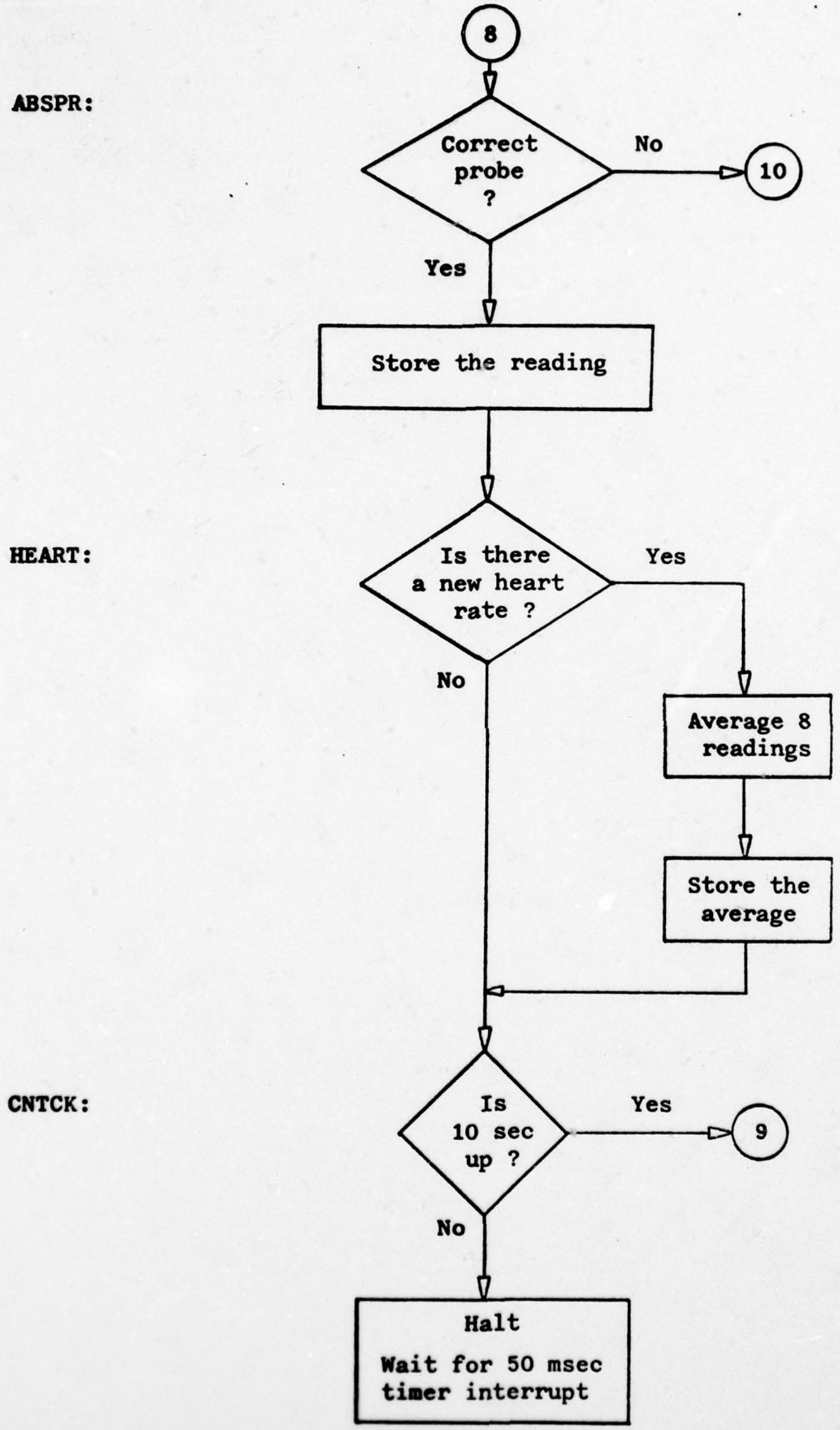


Fig. 19. Operating System Flow Chart (Sheet 8 of 9)

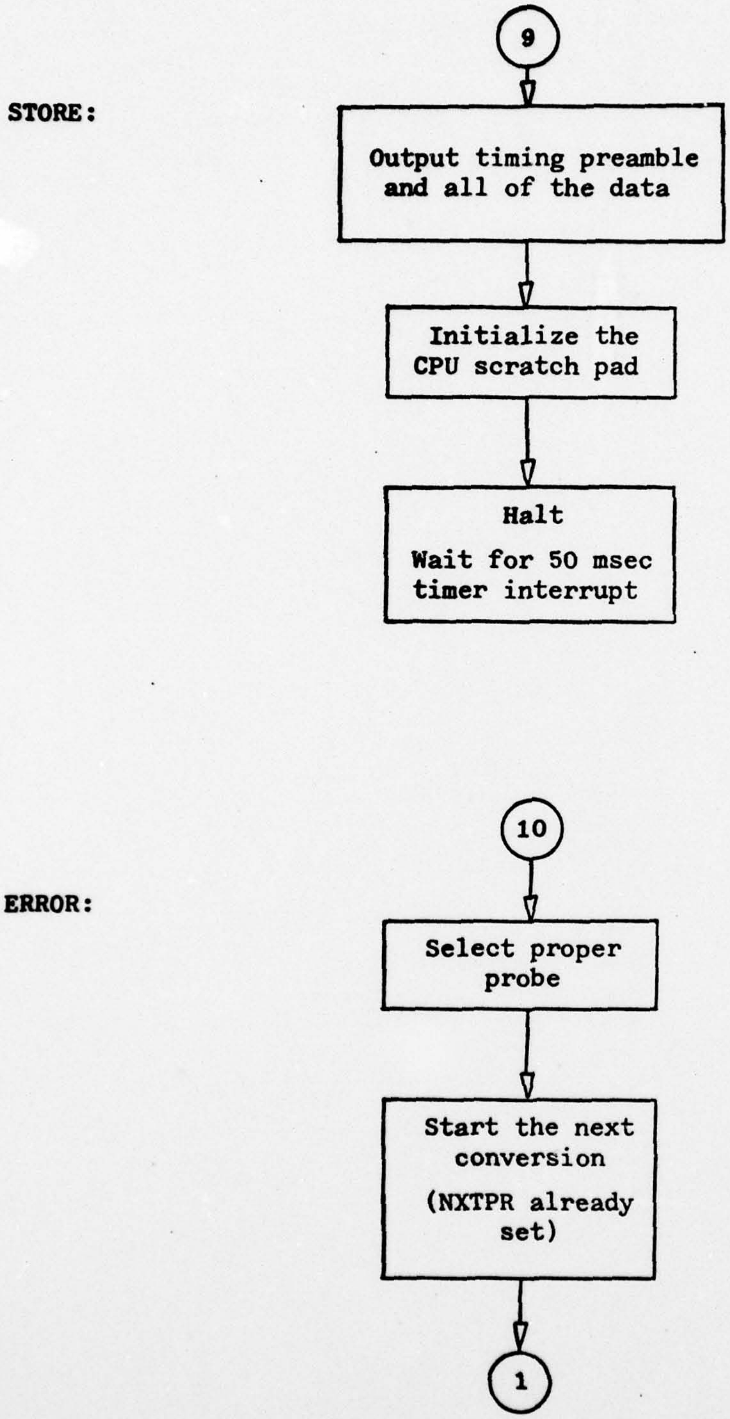


Fig. 19. Operating System Flow Chart (Sheet 9 of 9)

```
**************************************************
*                                                *
*                                                *
*           IFPDAS II PROTOTYPE                  *
*             OPERATING SYSTEM                   *
*                                                *
*                                                *
**************************************************
*                                                *
*                                                *
*        THIS PROGRAM CONTROLS THE OPERATION     *
*   OF THE IFPDAS II PROTOTYPE HARDWARE.         *
*   EXECUTION OF THE OPERATING SYSTEM BY THE     *
*   CPU RESULTS IN INITIALIZATION OF THE         *
*   SYSTEM AND ACQUISITION, REDUCTION, AND       *
*   STORAGE OF THE DATA.  THE OPERATING          *
*   SYSTEM SELECTS THE PROPER INPUT SENSOR,      *
*   STARTS THE A/D CONVERSION, AND INPUTS        *
*   THE DATA FOR REDUCTION AND PERMANENT         *
*   STORAGE.  IN ADDITION, IT ENSURES THAT       *
*   THE PROPER SERVICE ROUTINE RECEIVES          *
*   THE DATA.                                    *
*                                                *
*                                                *
**************************************************
*                                                *
*                                                *
*        THE OPERATING SYSTEM CONSISTS OF        *
*   THE FOLLOWING PROGRAM MODULES:               *
*                                                *
*        PWRUP - CONFIGURES THE IFPDAS II        *
*                HARDWARE FOR THE DATA           *
*                ACQUISITION FUNCTION            *
*                                                *
*        T50MS - RESETS THE 50 MSEC TIMER        *
*                AND STARTS THE PROGRAM          *
*                LOOP                            *
*                                                *
*        EOC - OBTAINS THE DATA AND PASSES       *
*              IT TO THE APPROPRIATE             *
*              SERVICE ROUTINE                   *
*                                                *
*        PO2IN - INSPIRED OXYGEN PARTIAL         *
*                PRESSURE SERVICE ROUTINE        *
*                                                *
*        PO2OUT - EXPIRED OXYGEN PARTIAL         *
*                 PRESSURE SERVICE ROUTINE       *
*                                                *
*        FLWRT - FLOW RATE SERVICE ROUTINE       *
*                                                *
*        BRRT - BREATHING RATE CALCULATIONS      *
```

```
;  *                                                 *
;  *        GX - X-G'S SERVICE ROUTINE               *
;  *                                                 *
;  *        GY - Y-G'S SERVICE ROUTINE               *
;  *                                                 *
;  *        GZ - Z-G'S SERVICE ROUTINE               *
;  *                                                 *
;  *        ABSPR - ABSOLUTE PRESSURE                *
;  *                SERVICE ROUTINE                  *
;  *                                                 *
;  *        HEART - HEART RATE CALCULATIONS          *
;  *                                                 *
;  *        CNTCK - TEN SECOND COUNTER               *
;  *                                                 *
;  *        STORE - OUTPUTS THE DATA EVERY           *
;  *                10 SECONDS                       *
;  *                                                 *
;  *        ERROR - CORRECTS SERVICE ROUTINE         *
;  *                PROBE MISMATCHES                 *
;  *                                                 *
;  *                                                 *
;  *        THE OPERATING SYSTEM INCLUDES THE        *
;  *   FOLLOWING SUBROUTINES:                        *
;  *                                                 *
;  *        STRTAD - CALLS STROBE AND TRIGGR         *
;  *                                                 *
;  *        STROBE - INCREMENTS THE DAS 1128         *
;  *                 MUX ADDR                        *
;  *                                                 *
;  *        TRIGGR - STARTS THE A/D CONVERSION       *
;  *                                                 *
;  *        SUM - PERFORMS DOUBLE PRECISION          *
;  *              ADDITION                           *
;  *                                                 *
;  *        AVG - PERFORMS DATA AVERAGING            *
;  *                                                 *
;  *        INIT - INITIALIZES CPU SCRATCH PAD       *
;  *                                                 *
;  *        DTOUT - PREPARES DATA FOR OUTPUT AND     *
;  *                CALLS ASCII AND PRINT            *
;  *                                                 *
;  *        ASCII - CONVERTS BINARY DATA TO ASCII    *
;  *                                                 *
;  *        PRINT - OUTPUTS ASCII CHARACTER          *
;  *                TO THE CONSOLE                   *
;  *                                                 *
;  *                                                 *
;  ***************************************************
```

```
                  ;   ********************************************
                  ;   *                                          *
                  ;   *                                          *
                  ;   *         ASSEMBLER EQU STATEMENTS         *
                  ;   *                                          *
                  ;   *                                          *
                  ;   ********************************************
                  ;
                  ;
004E              MODE    EQU    04EH      ;MODE SET FOR USART INIT
0037              RSTURT  EQU    037H      ;CMD INST TO RESET USART
00ED              USART   EQU    0EDH      ;USART CONTROL PORT
                  ;
0001              READY   EQU    01H       ;MASK FOR TRANSMITTER READY
00EC              CON     EQU    0ECH      ;CONSOLE OUTPUT PORT
                  ;
3F80              STACK   EQU    3F80H     ;INITIAL STACK POINTER VALU
                  ;
00E4              DATA    EQU    0E4H      ;MUX ADDR INPUT PORT
                                           ;   (PORT A OF 8255 #1)
00E5              MUX     EQU    0E5H      ;CONVERTED DATA INPUT PORT
                                           ;   (PORT B OF 8255 #1)
00E6              STAT1   EQU    0E6H      ;STATUS PORT OF 8255 #1
00E7              PPI1    EQU    0E7H      ;8255 #1 CONTROL PORT
00E8              HRIN    EQU    0E8H      ;HEART RATE INPUT PORT
                                           ;   (PORT A OF 8255 #2)
00EA              STAT2   EQU    0EAH      ;STATUS PORT OF 8255 #2
00EB              PPI2    EQU    0EBH      ;8255 #2 CONTROL PORT
00B6              AIBI    EQU    0B6H      ;8255 CONTROL WORD:
                                           ;   A - MODE 1, INPUT
                                           ;   B - MODE 1, INPUT
00B4              AIBO    EQU    0B4H      ;8255 CONTROL WORD:
                                           ;   A - MODE 1, INPUT
                                           ;   B - MODE 1, OUTPUT
000C              C6OFF   EQU    0CH       ;TURNS OFF C6
000E              C7OFF   EQU    0EH       ;TURNS OFF C7
0020              ABFMSK  EQU    020H      ;MASK TO OBTAIN IBFA
0002              BBFMSK  EQU    002H      ;MASK TO OBTAIN OBFB
                  ;
000D              STBON   EQU    0DH       ;TURNS ON STROBE (SETS C6)
000C              STBOFF  EQU    0CH       ;TURNS OFF STROBE (C6 OFF)
000F              TRGON   EQU    0FH       ;TURNS ON TRIGGER (SETS C7)
000E              TRGOFF  EQU    0EH       ;TURNS OFF TRIGGER (C7 OFF)
00E7              ADCON   EQU    0E7H      ;PSEUDO ADDR FOR DAS 1128
                                           ;     VIA PORT C OF 8255 #1
                  ;
0036              C0M3    EQU    036H      ;CTR 0 TO MODE 3
0070              C1M0    EQU    070H      ;CTR 1 TO MODE 0
00B6              C2M3    EQU    0B6H      ;CTR 2 TO MODE 3
00DC              CTR0    EQU    0DCH      ;COUNTER 0 PORT
00DD              CTR1    EQU    0DDH      ;COUNTER 1 PORT
00DE              CTR2    EQU    0DEH      ;COUNTER 2 PORT
00DF              TMCP    EQU    0DFH      ;TIMER COMMAND PORT
```

```
00F6            ICW1     EQU      0F6H       ;INTERRUPT CMD WORD 1
0003            ICW2     EQU      003H       ;INTERRUPT CMD WORD 2
00DA            ICCP1    EQU      0DAH       ;INT CONTROLLER CMD PORT 1
00DB            ICCP2    EQU      0DBH       ;INT CONTROLLER CMD PORT 2
0004            IMASK    EQU      04H        ;INT MASK VALUE
                                             ;   MASKS OFF LEVEL 2
00DB            MSKPT    EQU      0DBH       ;INT MASK PORT
                ;
0020            TRHLD    EQU      20H        ;BREATHING RATE THRESHOLD
00FF            FLAG     EQU      0FFH       ;SET FLAG VALUE
                ;
00F0            MUXMSK   EQU      0F0H       ;MASK FOR MUX ADDR
0060            PRBCNT   EQU      60H        ;PROBE COUNT (#0 - 6)
0020            EOIC     EQU      020H       ;END OF INT CMD WRD
0011            BYTES    EQU      017D       ;# OF OUTPUT BYTES
000B            LOCS     EQU      11D        ;# OF LOCS TO BE ZEROED
                ;
007E            CSP      EQU      7EH        ;ASCII CODE FOR 'CS.'
001E            PR       EQU      1EH        ;ASCII CODE FOR 'PRINT'
```

```
;  *************************************************
;  *                                               *
;  *                                               *
;  *          POWER UP ROUTINE                     *
;  *                                               *
;  *                                               *
;  *************************************************
;  *                                               *
;  *                                               *
;  *      THE POWER-UP ROUTINE IS EXECUTED WHEN    *
;  *  THE HARDWARE RECOGNIZES THAT POWER HAS       *
;  *  BEEN APPLIED TO THE IFPDAS.  (THE POWER-     *
;  *  UP RESET TRIGGERS OFF THE +5 VOLT SUPPLY.)   *
;  *  THE 8251 COMMUNICATION INTERFACE (USART),    *
;  *  8255 I/O PORTS, 8253 TIMERS, 8259            *
;  *  INTERRUPT CONTROLLER, DAS 1128 DATA          *
;  *  ACQUISITION MODULE, AND THE CPU SCRATCH      *
;  *  PAD STORAGE AREA ARE ALL INITIALIZED.        *
;  *                                               *
;  *      THE POWER-UP ROUTINE STARTS THE          *
;  *  50 MSEC TIMER; WHEN IT TIMES OUT, THE        *
;  *  PROGRAM LOOP IS ENTERED AND THE DATA         *
;  *  ACQUISITION FUNCTION IS BEGUN.               *
;  *                                               *
;  *                                               *
;  *************************************************
;
;
;
;
```

```
                        :  ***********************************************
                        :  *                                             *
                        :  *                                             *
                        :  *       USART INITIALIZATION                  *
                        :  *                                             *
                        :  *                                             *
                        :  *            MODE 4E : 1 STOP BIT             *
                        :  *                      NO PARITY              *
                        :  *                      8 BITS                 *
                        :  *                      16X BAUD FACTOR        *
                        :  *                                             *
                        :  *            CMD  37 : NO HUNT MODE           *
                        :  *                      NO INTERNAL RESET      *
                        :  *                      RTS HIGH               *
                        :  *                      RESET ERROR FLAGS      *
                        :  *                      NORMAL BREAK CHAR      *
                        :  *                      RECEIVE ENABLED        *
                        :  *                      DTR HIGH               *
                        :  *                      TRANSMIT ENABLED       *
                        :  *                                             *
                        :  *                                             *
                        :  *       STACK POINTER INITIALIZATION          *
                        :  *                                             *
                        :  *                 SP STARTS AT 3F80H          *
                        :  *                                             *
                        :  *                                             *
                        :  ***********************************************
                        :
                        :
0000                            ORG     0
0000    3E4E                    MVI     A,MODE      :OUTPUT MODE SET TO USART
0002    D3ED                    OUT     USART
0004    3E37                    MVI     A,RSTURT    :OUTPUT RESET COMMAND WORD
0006    D3ED                    OUT     USART       :      TO USART
0008    31803F                  LXI     SP,STACK    :INITIALIZES STACK POINTER
```

```
                            ;
                            ;
                            ;   ****************************************
                            ;   *                                      *
                            ;   *                                      *
                            ;   *         8255 INITIALIZATION          *
                            ;   *                                      *
                            ;   *                                      *
                            ;   *           8255 #1                    *
                            ;   *              MODE 1 (STROBED I/O)   B6 :   *
                            ;   *                       SET MODE       *
                            ;   *                       PORT A - MODE 1 *
                            ;   *                       PORT A - INPUT *
                            ;   *                       C6,C7 - OUTPUT *
                            ;   *                       PORT B - MODE 1 *
                            ;   *                       PORT B - INPUT *
                            ;   *              BITS C6 & C7 OFF        *
                            ;   *                                      *
                            ;   *           8255 #2                    *
                            ;   *              MODE 1 (STROBED I/O) B4 :  *
                            ;   *                       SET MODE       *
                            ;   *                       PORT A - MODE 1 *
                            ;   *                       PORT A - INPUT *
                            ;   *                       C6,C7 - OUTPUT *
                            ;   *                       PORT B - MODE 1 *
                            ;   *                       PORT B - OUTPUT *
                            ;   *                                      *
                            ;   *                                      *
                            ;   ****************************************
                            ;
                            ;
000B     3EB6                   MVI      A,AIBI      ;OUTPUT CONTROL WORD
000D     D3E7                   OUT      PPI1        ;     TO 8255 #1
000F     3E0C                   MVI      A,C6OFF     ;ENSURE BIT C6 IS OFF
0011     D3E7                   OUT      PPI1
0013     3E0E                   MVI      A,C7OFF     ;ENSURE BIT C7 IS OFF
0015     D3E7                   OUT      PPI1
0017     3EB4                   MVI      A,AIBO      ;OUTPUT CONTROL WORD
0019     D3EB                   OUT      PPI2        ;   TO 8255 #2
                            ;
                            ;
                            ;   ****************************************
                            ;   *                                      *
                            ;   *                                      *
                            ;   *      INTERRUPT INITIALIZATION        *
                            ;   *                                      *
                            ;   *                                      *
                            ;   *          ICW1  F6  :  JUMP TABLE AT  *
                            ;   *          ICW2  03  :  03E0 TO 03FF   *
                            ;   *                       INTERVAL = 4   *
                            ;   *          IMASK 04  :  MASK OFF LEVEL 2 *
                            ;   *                                      *
                            ;   *                                      *
                            ;   ****************************************
                            ;
                            ;
001B     3EF6                   MVI      A,ICW1
001D     D3DA                   OUT      ICCP1       ;OUTPUT COMMAND WORD 1
001F     3E03                   MVI      A,ICW2
0021     D3DB                   OUT      ICCP2       ;OUTPUT COMMAND WORD 2
0023     3E04                   MVI      A,IMASK
0025     D3DB                   OUT      MSKPT       ;OUTPUT MASK WORD
```

```
                        ;
                        ;
                        ;    ************************************************
                        ;    *                                              *
                        ;    *                                              *
                        ;    *      TIMER INITIALIZATION                    *
                        ;    *                                              *
                        ;    *                                              *
                        ;    *          CONTROL WORD 36 :                   *
                        ;    *                  SELECT COUNTER 0            *
                        ;    *                  LOAD 2 BYTES                *
                        ;    *                  MODE 3                      *
                        ;    *                  BINARY COUNTER              *
                        ;    *                                              *
                        ;    *          CONTROL WORD 70 :                   *
                        ;    *                  SELECT COUNTER 1            *
                        ;    *                  LOAD 2 BYTES                *
                        ;    *                  MODE 0                      *
                        ;    *                  BINARY COUNTER              *
                        ;    *                                              *
                        ;    *          CONTROL WORD 86 :                   *
                        ;    *                  SELECT COUNTER 2            *
                        ;    *                  LOAD 2 BYTES                *
                        ;    *                  MODE 3                      *
                        ;    *                  BINARY COUNTER              *
                        ;    *                                              *
                        ;    *                                              *
                        ;    *          SQUARE WAVE GEN SET TO 225 HZ       *
                        ;    *                                              *
                        ;    *          BAUD RATE SET TO 1200               *
                        ;    *                                              *
                        ;    *                                              *
                        ;    ************************************************
                        ;
                        ;
0027    3E36            MVI     A,C0M3      ;INIT COUNTER 0 FOR SQUARE
0029    D3DF            OUT     TMCP        ;   WAVE GENERATOR
002B    3EAB            MVI     A,0ABH
002D    D3DC            OUT     CTR0        ;ESTABLISHES 225 HZ CLOCK
002F    3E12            MVI     A,012H      ;   FOR HEART RATE CIRCUIT
0031    D3DC            OUT     CTR0
0033    3E70            MVI     A,C1M0      ;RESET TIMER 1
0035    D3DF            OUT     TMCP
0037    3EB6            MVI     A,C2M3      ;INIT COUNTER 2 FOR BAUD
0039    D3DF            OUT     TMCP        ;   RATE
003B    3E38            MVI     A,56D
003D    D3DE            OUT     CTR2        ;ESTABLISHES 1200 BAUD RATE
003F    AF              XRA     A           ;   FOR HAZELTINE CRT
0040    D3DE            OUT     CTR2
```

```
                          :  ***********************************************
                          :  *                                             *
                          :  *                                             *
                          :  *            DAS 1128 INITIALIZATION          *
                          :  *                                             *
                          :  *                                             *
                          :  *                 SEQUENTIALLY TRIGGERED      *
                          :  *                                             *
                          :  *                 ONE CONVERSION              *
                          :  *                 READ MUX ADDRESS INFO       *
                          :  *                 CONVERT UNTIL MUX ADDR      *
                          :  *                    EQUALS PRBCNT            *
                          :  *                                             *
                          :  *                                             *
                          :  ***********************************************
                          :
                          :
0042    CDC002            CALL    TRIGGR      :STARTS A CONVERSION
0045    DBE5      PWRUP:  IN      STAT1       :INPUT STATUS OF 8255 #1
0047    E602              ANI     BBFMSK      :MASKS IBFB
0049    CA4500            JZ      PWRUP       :LOOP IF STILL CONVERTING
004C    DBE5              IN      MUX         :READS IN MUX ADDR & 4 LSB
004E    E6F0              ANI     MUXMSK      :SAVES B7-B4
0050    FE50              CPI     PRBCNT      :COMPARE MUX ADDR TO LAST
                                              :    ADDR (INFO IN BITS 7-4)
0052    CA5900            JZ      IDA         :JUMP IF MUX ADDR = PRBCNT
0055    CD8002            CALL    STRTAD      :START A CONVERSION
0058    C34500            JMP     PWRUP
```

```
                    ;   ****************************************************
                    ;   *                                                  *
                    ;   *                                                  *
                    ;   *         THIS SECTION INITIALIZES THE             *
                    ;   *            STORAGE AREA.  TIME, PINSM            *
                    ;   *            POUTS, HRTRT, GXMAX, GYMAX, &         *
                    ;   *            GZMAX ARE ZEROED.  THEN OFFH          *
                    ;   *            IS LOADED INTO GXMIN, GYMIN,          *
                    ;   *            & GZMIN.  200D IS LOADED              *
                    ;   *            INTO THE TEN SEC COUNTER.             *
                    ;   *                                                  *
                    ;   *                                                  *
                    ;   ****************************************************
                    ;
                    ;
005B    AF          IDA:    XRA     A
005C    21003C              LXI     H,TIME      ;ZERO TIME COUNT BYTE 1
005F    77                  MOV     M,A
0060    23                  INX     H           ;ZERO TIME COUNT BYTE 2
0061    77                  MOV     M,A
0062    CDE702              CALL    INIT        ;INITIALIZE PINSM TO GZMIN
0065    21133C              LXI     H,TENSC
0068    36C8                MVI     M,200D      ;SETS THE COUNTER TO 200
006A    AF                  XRA     A
006B    160F                MVI     D,15
006D    23          IDA1:   INX     H           ;ZEROES THE EVENT COUNTERS
006E    77                  MOV     M,A         ;   AND TEMPORARY RUNNING
006F    15                  DCR     D           ;   SUMS
0070    C26D00              JNZ     IDA1
                    ;
                    ;
                    ;   ****************************************************
                    ;   *                                                  *
                    ;   *                                                  *
                    ;   *         THIS SECTION STARTS THE 50 MS            *
                    ;   *            TIMER.  WHEN THE TIMER EXPIRES        *
                    ;   *            THE PROGRAM LOOP IS ENTERED           *
                    ;   *            AT T50MS.                             *
                    ;   *                                                  *
                    ;   *                                                  *
                    ;   ****************************************************
                    ;
                    ;
0073    3EE5                MVI     A,0E5H
0075    D30D                OUT     CTR1        ;MOVES E5 TO LSB OF COUNTER
0077    3ED1                MVI     A,0D1H
0079    D30D                OUT     CTR1        ;MOVES D1 TO MSB OF COUNTER
007B    FB                  EI
007C    76                  HLT                 ;WAIT FOR 50 MS TIMER
007D    00                  NOP
```

PROGRAM LOOP

THE PROGRAM LOOP IS EXECUTED WHEN THE 50 MSEC TIMER EXPIRES. IT CONTAINS THE SERVICE ROUTINES - ONE FOR EACH PHYSIOLOGICAL AND ENVIRONMENTAL SENSOR. THE DATA IS COLLECTED BY THE END-OF-CONVERSION (EOC) ROUTINE WHICH AVERAGES 6 SAMPLES INTO 1 READING WHICH IT PASSES TO THE APPROPRIATE SUBROUTINE.

THE "END-OF-CONVERSION" IS DETECTED BY CHECKING THE "INPUT-BUFFER-FULL" (IBF) STATUS BIT. THIS METHOD WAS USED, INSTEAD OF HAVING THE 8255 INTERRUPT THE CPU, TO ELIMINATE THE OVERHEAD TO HAVE THE CPU HANDLE THE INTERRUPT. THE TIME TO COMPLETE THE CONVERSION IS KNOWN, 25 MICRO-SEC: AND, NORMALLY, THE TIME FROM THE TRIGGER TO THE CHECK OF THE IBF BIT IS MORE THAN 25 MICRO-SEC.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
                        ;
                        ;
                        ;
                        ;   *************************************************
                        ;   *                                               *
                        ;   *                                               *
                        ;   *           TIMER INTERRUPT HANDLER             *
                        ;   *                                               *
                        ;   *                                               *
                        ;   *           THIS CODE IS EXECUTED WHEN THE      *
                        ;   *              50 MS TIMER EXPIRES              *
                        ;   *                                               *
                        ;   *                                               *
                        ;   *           THE TIMER IS RESET AND THE          *
                        ;   *              FIRST CONVERSION (PO2 IN)        *
                        ;   *              IS STARTED                       *
                        ;   *                                               *
                        ;   *           NXTPR  IS USED BY THE END OF        *
                        ;   *              CONVERSION (EOC) ROUTINE         *
                        ;   *              TO DETERMINE WHICH PROBE         *
                        ;   *              WAS SAMPLED.  THE ADDR OF        *
                        ;   *              THE INDIVIDUAL PROBE SERVICE     *
                        ;   *              ROUTINE IS STORED IN NXTPR.      *
                        ;   *                                               *
                        ;   *                                               *
                        ;   *************************************************
                        ;
                        ;
007E    3E70    T50MS:  MVI     A,C1MD      ;RESETS COUNTER 1
0080    D30F            OUT     TMCP
0082    3EE5            MVI     A,0E5H      ;MOVES E5 TO LSB OF COUNTER
0084    D30D            OUT     CTR1
0086    3E01            MVI     A,001H      ;MOVES 01 TO MSB OF COUNTER
0088    D30D            OUT     CTR1
008A    3E20            MVI     A,EOIC      ;RESETS "IN SERVICE" BIT
008C    D30A            OUT     ICCP1
008E    CD8002          CALL    STRTAD      ;INCR THE MUX ADDR (TO 00)
                                            ;   AND START A CONVERSION
0091    21C300          LXI     H,PO2IN     ;GET ADDR OF "PO2 IN"
                                            ;   SERVICE ROUTINE
0094    22113C          SHLD    NXTPR       ;STORE ADDR IN NXTPR
0097    33              INX     SP          ;RESTORES SP TO ORIGINAL
0098    33              INX     SP          ;   VALUE
0099    C39300          JMP     EOC
```

AD-A055 638 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 6/16
AIRCREW INFLIGHT PHYSIOLOGICAL DATA ACQUISITION SYSTEM II.(U)
DEC 77 J G JOLDA, S J WANZEK
UNCLASSIFIED AFIT/GE/EE-77-21 NL

2 OF 2
ADA
055638

END
DATE
FILMED
8-78
DDC

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
                  ;   **************************************************
                  ;   *                                                *
                  ;   *                                                *
                  ;   *          END-OF-CONVERSION (EOC) ROUTINE       *
                  ;   *                                                *
                  ;   *                                                *
                  ;   *          THIS CODE GENERATES EIGHT             *
                  ;   *             CONVERSIONS WHICH ARE SUMMED       *
                  ;   *             AND DIVIDED BY 8 TO OBTAIN         *
                  ;   *             AN AVERAGE READING - THUS          *
                  ;   *             REDUCING THE NOISE EFFECT.         *
                  ;   *                                                *
                  ;   *          THE SUM IS STORED IN H&L.  D          *
                  ;   *             CONTAINS THE COUNT OF THE          *
                  ;   *             CONVERSIONS.  THE AVERAGE          *
                  ;   *             READING IS PASSED TO THE           *
                  ;   *             SERVICE ROUTINES IN E.             *
                  ;   *                                                *
                  ;   *          AFTER THE AVERAGE HAS BEEN COM-       *
                  ;   *             PUTED, THE MUX ADDR IS             *
                  ;   *             OBTAINED AND MASKED OFF            *
                  ;   *                                                *
                  ;   *          THEN A JUMP INDIRECT IS ACCOMP-       *
                  ;   *             LISHED TO THE SERVICE ROUTINE      *
                  ;   *             ADDR STORED IN NXTPR               *
                  ;   *                                                *
                  ;   *                                                *
                  ;   **************************************************
                  ;
                  ;
009C   210000     EOC:      LXI     H,0         ;ZEROES H&L FOR SUM
009F   1608                 MVI     D,8         ;SET COUNTER
00A1   DBE5       EOC1:     IN      STAT1       ;INPUT STATUS OF 8255 #1
00A3   E602                 ANI     OBFMSK      ;MASKS IBFB
00A5   CAA100               JZ      EOC1        ;JUMP IF STILL CONVERTING
00A8   15                   DCR     D           ;DECREMENT CONV COUNTER
00A9   CAB800               JZ      EOC2
00AC   CDC302               CALL    TRIGGR      ;START THE NEXT CONVERSION
00AF   DBE4                 IN      DATA        ;INPUT THE CONVERTED DATA
00B1   2F                   CMA                 ;COMPLEMENTS DATA FROM
                                                ;   INVERTING DRIVER (8226)
00B2   CDC902               CALL    SUM         ;ADD DATA TO RUNNING SUM
00B5   C3A100               JMP     EOC1
00B8   DBE4       EOC2:     IN      DATA        ;INPUT THE 8TH READING
00BA   2F                   CMA                 ;COMPLEMENTS DATA FROM
                                                ;   INVERTING DRIVER (8226)
00BB   CDC902               CALL    SUM         ;ADD TO RUNNING SUM
00BE   CDCE02               CALL    AVG         ;AVERAGE THE RUNNING SUM
00C1   DBE5                 IN      MUX         ;READS IN MUX ADDR & 4 LSB
00C3   E6F0                 ANI     MUXMSK      ;SAVES B7-B4
00C5   2A1130               LHLD    NXTPR       ;BRANCH INDIRECT TO THE
00C8   E9                   PCHL                ;   ADDR IN NXTPR
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
                      ;   ***********************************************
                      ;   *                                             *
                      ;   *                                             *
                      ;   *               THIS IS THE SERVICE ROUTINE   *
                      ;   *                  FOR PO2 IN                 *
                      ;   *                                             *
                      ;   *                                             *
                      ;   *               THE NEXT CONVERSION IS STARTED*
                      ;   *                 (PO2 OUT).  THE ADDR OF PO2OUT*
                      ;   *                  IS STORED IN NXTPR .  THE  *
                      ;   *                  CURRENT READING OF PO2IN IS*
                      ;   *                  INPUT, THEN ADDED TO THE   *
                      ;   *                  PREVIOUS SUM, AND STORED   *
                      ;   *                  IN PINSM .                 *
                      ;   *                                             *
                      ;   *               DOUBLE PRECISION ADDITION     *
                      ;   *                 (DAD) IS USED               *
                      ;   *                                             *
                      ;   *                                             *
                      ;   *               THE A REG CONTAINS THE MUX ADDR*
                      ;   *                  IN BITS 7-4 (X0H)          *
                      ;   *                                             *
                      ;   *                                             *
                      ;   ***********************************************
                      ;
                      ;
00C9   0600    PO2IN:  MVI    B,00H     ;PUT DESIRED MUX ADDR IN B
00CB   B8              CMP    B         ;IS THE MUX ADDR 0 ?
00CC   C29902          JNZ    ERROR     ;IF NOT THERE IS AN ERROR
00CF   CD9002          CALL   STRTAD    ;START THE NEXT CONVERSION
00D2   21E200          LXI    H,PO2OUT  ;GET ADDR OF "PO2 OUT"
                                        ;    SERVICE ROUTINE
00D5   22113C          SHLD   NXTPR     ;STORE ADDR IN NXTPR
00D8   2A023C          LHLD   PINSM     ;PLACE THE PO2IN RUNNING
                                        ;    SUM IN H&L
                                        ;NOTE: E HOLDS THE AVG DATA
                                        ;       D IS ZERO
00DB   19              DAD    D         ;H&L = H&L + D&E
00DC   22023C          SHLD   PINSM     ;STORES NEW SUM IN PINSM
00DF   C39000          JMP    EOC
```

```
                        ;  ***********************************************
                        ;  *                                             *
                        ;  *                                             *
                        ;  *          THIS IS THE SERVICE ROUTINE        *
                        ;  *             FOR PO2 OUT                     *
                        ;  *                                             *
                        ;  *                                             *
                        ;  *          THE NEXT CONVERSION IS STARTED     *
                        ;  *             (FLOW RATE).  THE ADDR OF       *
                        ;  *             FLWRT IS STORED IN NXTPR.       *
                        ;  *          THE CURRENT READING OF PO2OUT      *
                        ;  *             IS INPUT, THEN ADDED TO         *
                        ;  *             THE PREVIOUS SUM, AND STORED    *
                        ;  *             IN POUTS.                       *
                        ;  *                                             *
                        ;  *          DOUBLE PRECISION ADDITION          *
                        ;  *             (DAD) IS USED                   *
                        ;  *                                             *
                        ;  *                                             *
                        ;  *          THE A REG CONTAINS THE MUX ADDR    *
                        ;  *             IN BITS 7-4 (X0H).              *
                        ;  *                                             *
                        ;  *                                             *
                        ;  ***********************************************
                        ;
                        ;
00E2    0610    PO2OUT: MVI     B,10H     ;PUT DESIRED MUX ADDR IN B
00E4    B8              CMP     B         ;IS THE MUX ADDR 1 ?
00E5    C29302          JNZ     ERROR     ;IF NOT, THERE IS AN ERROR
00E8    CD9302          CALL    STRTAD    ;START THE NEXT CONVERSION
00EB    21F300          LXI     H,FLWRT   ;GET ADDR OF "FLOW RATE"
                                          ;   SERVICE ROUTINE
00EE    22113C          SHLD    NXTPR     ;STORE ADDR IN NXTPR
00F1    2A043C          LHLD    POUTS     ;PLACE THE PO2OUT RUNNING
                                          ;   SUM IN H&L
                                          ;NOTE: E HOLDS THE AVG DATA
                                          ;      D IS ZERO
00F4    19              DAD     D         ;H&L = H&L + D&E
00F5    22043C          SHLD    POUTS     ;STORES NEW SUM IN POUTS
00F8    C39300          JMP     EOC
```

```
                            :    ********************************************
                            :    *                                          *
                            :    *                                          *
                            :    *         THIS IS THE SERVICE ROUTINE      *
                            :    *            FOR FLOW RATE                 *
                            :    *                                          *
                            :    *                                          *
                            :    *         THE NEXT CONVERSION IS STARTED   *
                            :    *         THE ADDR OF GX IS STORED IN NXTPR *
                            :    *         THE CURRENT READING OF FLOW RATE *
                            :    *            IS INPUT, THEN ADDED TO THE   *
                            :    *            PREVIOUS SUM, AND STORED IN   *
                            :    *            FLRTS                         *
                            :    *                                          *
                            :    *         DOUBLE PRECISION ADDITION (DAD)  *
                            :    *            IS USED                       *
                            :    *                                          *
                            :    *         THE A REG CONTAINS THE MUX ADDR  *
                            :    *            IN BITS 7-4 (X0H)             *
                            :    *                                          *
                            :    *                                          *
                            :    ********************************************
                            :
                            :
00FB    0620        FLWRT:  MVI     B,20H       :PUT DESIRED MUX ADDR IN B
00FD    B8                  CMP     B           :IS THE MUX ADDR 2 ?
00FE    C29302              JNZ     ERROR       :IF NOT, THERE IS AN ERROR
0101    CD3002              CALL    STRTAD      :START THE NEXT CONVERSION
0104    214001              LXI     H,GX        :GET ADDR OF "X-G'S"
                                                :   SERVICE ROUTINE
0107    22113C              SHLD    NXTPR       :STORE THE ADDR IN NXTPR
010A    2A063C              LHLD    FLRTS       :PLACE THE FLOWRATE RUNNING
                                                :   SUM IN H&L
                                                :NOTE: E HOLDS THE AVG DATA
                                                :      D IS ZERO
010D    19                  DAD     D           :H&L = H&L + D&E
010E    22063C              SHLD    FLRTS       :STORES NEW SUM IN FLRTS
```

```
                           :       *****************************************************
                           :       *                                                   *
                           :       *                                                   *
                           :       *           THIS SECTION USES THE CURRENT           *
                           :       *              READING OF FLOW RATE TO              *
                           :       *              DETECT A NEW BREATH AND              *
                           :       *              COUNT THE NUMBER OF INTERVALS        *
                           :       *              BETWEEN BREATHS.                     *
                           :       *                                                   *
                           :       *           THE NEW BREATH IS DETECTED WHEN         *
                           :       *              THE FLOW RATE READING IS FOUND       *
                           :       *              TO BE BELOW THE THRESHOLD 4          *
                           :       *              TIMES IN A ROW.                      *
                           :       *                                                   *
                           :       *           THE NUMBER OF 50 MS INTERVALS           *
                           :       *              BETWEEN BREATHS IS STORED            *
                           :       *              IN BTHRT.                            *
                           :       *                                                   *
                           :       *           E HOLDS THE FLOW RATE READING           *
                           :       *           TFLAG HOLDS THE THRESHOLD FLAG          *
                           :       *           THCNT HOLDS THE THRESHOLD COUNT         *
                           :       *           TMCNT HOLDS THE 50 MS COUNT             *
                           :       *                                                   *
                           :       *                                                   *
                           :       *****************************************************
                           :
                           :
                           :
0111    21173C    BRRT:    LXI    H,TMCNT    :GET ADDR OF 50 MS COUNT
0114    34                 INR    M          :INCREMENT THE COUNT
0115    7B                 MOV    A,E        :MOVE CURRENT READING TO A
0116    FE20               CPI    TRHLD      :COMPARE TO THRESHOLD
0118    DA2501             JC     BRRT1      :JUMP IF BELOW THRESHOLD
0119    AF                 XRA    A
011C    32183C             STA    THCNT      :ZERO THE THRESHOLD COUNT
011F    32193C             STA    TFLAG      :   AND THRESHOLD FLAG
0122    C39C00             JMP    EOC        :DONE
0125    3A193C    BRRT1:   LDA    TFLAG      :GET THRESHOLD FLAG
0128    FEFF               CPI    FLAG       :IS IT SET?
012A    CA9C00             JZ     EOC        :DONE IF ALREADY SET
012D    3A183C             LDA    THCNT      :GET THRESHOLD COUNT
0130    3C                 INR    A          :INCREMENT THE COUNT
0131    32183C             STA    THCNT      :STORE NEW THRESHOLD CNT
0134    FE04               CPI    4
0136    C29C00             JNZ    EOC        :DONE IF .NE. 4
0139    3EFF               MVI    A,FLAG
013B    32193C             STA    TFLAG      :SET THRESHOLD FLAG
013E    7E                 MOV    A,M        :GET 50 MS COUNT
                                             :   (H&L SET TO TMCNT)
013F    21103C             LXI    H,BTHRT
0142    77                 MOV    M,A        :STORE COUNT FOR OUTPUT
0143    AF                 XRA    A
0144    32173C             STA    TMCNT      :ZERO THE 50 MS COUNT
0147    32183C             STA    THCNT      :   AND THRESHOLD COUNT
014A    C39C00             JMP    EOC
```

```
                          ;  ************************************************
                          ;  *                                              *
                          ;  *                                              *
                          ;  *          THIS IS THE SERVICE ROUTINE         *
                          ;  *             FOR X-G'S                        *
                          ;  *                                              *
                          ;  *                                              *
                          ;  *                                              *
                          ;  *          THE NEXT CONVERSION IS STARTED      *
                          ;  *          THE ADDR OF GY IS STORED IN NXTPR   *
                          ;  *          THE CURRENT READING OF X-G'S        *
                          ;  *             INPUT.  EIGHT READINGS ARE       *
                          ;  *             AVERAGED TO GIVE AN AVERAGE      *
                          ;  *             X-G'S READING.                   *
                          ;  *          IF THE AVG .GT. GXMAX IT IS         *
                          ;  *             IS STORED IN GXMAX - IF .LT.     *
                          ;  *             GXMIN IT IS STORED IN GXMIN      *
                          ;  *                                              *
                          ;  *          THE A REG CONTAINS THE MUX ADDR     *
                          ;  *            IN BITS 7-4 (X0H)                 *
                          ;  *                                              *
                          ;  *                                              *
                          ;  ************************************************
                          ;
                          ;
0140    0630     GX:      MVI     B,30H       ;PUT DESIRED MUX ADDR IN B
014F    B8                CMP     B           ;IS THE MUX ADDR 3 ?
0150    C29302            JNZ     ERROR       ;IF NOT, THERE IS AN ERROR
0153    CD8302            CALL    STRTAD      ;START THE NEXT CONVERSION
0156    219501            LXI     H,GY        ;GET ADDR OF "Y-G'S"
                                              ;   SERVICE ROUTINE
0159    22113C            SHLD    NXTPR       ;STORE ADDR IN NXTPR
015C    2A143C            LHLD    XG          ;LOADS H&L WITH X-G'S
                                              ;   RUNNING SUM
015F    7B                MOV     A,E         ;MOVES CURRENT READING TO A
0160    CDC302            CALL    SUM         ;ADD DATA TO RUNNING SUM
0163    3A103C            LDA     XGCNT       ;GET X-G COUNT
0166    3C                INR     A           ;INCREMENT THE COUNT
0167    FE08              CPI     8
0169    CA7501            JZ      GX1         ;JUMP IF XGCNT .EQ. 8
016C    22143C            SHLD    XG          ;STORE RUNNING SUM IN XG
016F    32103C            STA     XGCNT       ;STORE COUNT IN XGCNT
0172    C39300            JMP     EOC
0175    CDCE02   GX1:     CALL    AVG         ;AVERAGE THE 8 READINGS
0178    AF                XRA     A
0179    32103C            STA     XGCNT       ;ZERO THE COUNT AND
017C    32143C            STA     XG          ;   THE RUNNING SUM
017F    32153C            STA     XG+1
0182    7B                MOV     A,E         ;MOVES AVG DATA TO A
0183    21083C            LXI     H,GXMAX     ;GET ADDR OF MAX X-G'S
0186    BE                CMP     M           ;COMPARE AVG VALUE WITH MAX
0187    DA8B01            JC      GX2         ;JUMP IF A .LT. GXMAX
018A    77                MOV     M,A         ;REPLACE IF A.GE.GXMAX
018B    21093C   GX2:     LXI     H,GXMIN     ;GET ADDR OF MIN X-G'S
018E    BE                CMP     M           ;COMAPRE AVG VALUE WITH MIN
018F    D29300            JNC     EOC         ;JUMP IF A.GE.GXMIN
0192    77                MOV     M,A         ;REPLACE IF A.LT.GXMIN
0193    C39300            JMP     EOC
```

```
                        ;  ***********************************************
                        ;  *                                             *
                        ;  *                                             *
                        ;  *                                             *
                        ;  *          THIS IS THE SERVICE ROUTINE        *
                        ;  *             FOR Y-G'S                       *
                        ;  *                                             *
                        ;  *                                             *
                        ;  *          THE NEXT CONVERSION IS STARTED     *
                        ;  *             (Z-G'S)                         *
                        ;  *          THE ADDR OF GZ IS STORED IN NXTPR  *
                        ;  *          THE CURRENT READING OF Y-G'S IS    *
                        ;  *             INPUT.  EIGHT READINGS ARE      *
                        ;  *             AVERAGED TO GIVE AN AVERAGE     *
                        ;  *             Y-G'S READING.                  *
                        ;  *          IF THE AVG .GT. GYMAX IT IS        *
                        ;  *             STORED IN GYMAX - IF .LT.       *
                        ;  *             GYMIN IT IS STORED IN GYMIN     *
                        ;  *                                             *
                        ;  *          THE A REG CONTAINS THE MUX ADDR    *
                        ;  *             IN BITS 7-4 (X0H)               *
                        ;  *                                             *
                        ;  *                                             *
                        ;  ***********************************************
                        ;
                        ;
0196    0640      GY:        MVI     B,40H      ;PUT DESIRED MUX ADDR IN B
0198    B8                   CMP     B          ;IS THE MUX ADDR 4 ?
0199    C29902               JNZ     ERROR      ;IF NOT, THERE IS AN ERROR
019C    CD8002               CALL    STRTAD     ;START THE NEXT CONVERSION
019F    21DF01               LXI     H,GZ       ;GET ADDR OF "Z-G'S"
                                                ;   SERVICE ROUTINE
01A2    22113C               SHLD    NXTPR      ;STORE ADDR IN NXTPR
01A5    2A1D3C               LHLD    YG         ;LOADS H&L WITH Y-G'S
                                                ;   RUNNING SUM
01A8    7B                   MOV     A,E        ;MOVES CURRENT READING TO A
01A9    CDC302               CALL    SUM        ;ADD DATA TO RUNNING SUM
01AC    3A1F3C               LDA     YGCNT      ;GET Y-G COUNT
01AF    3C                   INR     A          ;INCREMENT THE COUNT
01B0    FE08                 CPI     8
01B2    CABE01               JZ      GY1        ;JUMP IF YGCNT .EQ. 8
01B5    221D3C               SHLD    YG         ;STORE RUNNING SUM IN YG
01B8    321F3C               STA     YGCNT      ;STORE COUNT IN YGCNT
01BB    C39C00               JMP     EOC
01BE    CDCE02    GY1:       CALL    AVG        ;AVERAGE THE 8 READINGS
01C1    AF                   XRA     A
01C2    321F3C               STA     YGCNT      ;ZERO THE COUNT AND
01C5    321D3C               STA     YG         ;   THE RUNNING SUM
01C8    321E3C               STA     YG+1
01CB    7B                   MOV     A,E        ;MOVES AVG DATA TO A
01CC    210A3C               LXI     H,GYMAX    ;GET ADDR OF MAX Y-G'S
01CF    BE                   CMP     M          ;COMPARE AVG VALUE WITH MAX
01D0    DAD401               JC      GY2        ;JUMP IF A.LT.GYMAX
01D3    77                   MOV     M,A        ;REPLACE IF A.GE.GYMAX
01D4    210B3C    GY2:       LXI     H,GYMIN    ;GET ADDR OF MIN Y-G'S
01D7    BE                   CMP     M          ;COMPARE AVG VALUE WITH MIN
01D8    D29C00               JNC     EOC        ;JUMP IF A.GE.GYMIN
01DB    77                   MOV     M,A        ;REPLACE IF A.LT.GYMIN
01DC    C39C00               JMP     EOC
```

```
                     :   ***********************************************
                     :   *                                             *
                     :   *                                             *
                     :   *         THIS IS THE SERVICE ROUTINE         *
                     :   *            FOR Z-G'S                        *
                     :   *                                             *
                     :   *                                             *
                     :   *                                             *
                     :   *         THE NEXT CONVERSION IS STARTED      *
                     :   *            (ABS PRESS)                      *
                     :   *         THE ADDR OF ABSPR IS STORED         *
                     :   *            IN NXTPR                         *
                     :   *         THE CURRENT READING OF Z-G'S IS     *
                     :   *            INPUT.  EIGHT READINGS ARE       *
                     :   *            AVERAGED TO GIVE AN AVERAGE      *
                     :   *            Z-G'S READING.                   *
                     :   *         IF THE AVG .GT. GZMAX IT IS         *
                     :   *            STORED IN GZMAX - IF .LT.        *
                     :   *            GZMIN IT IS STORED IN GZMIN      *
                     :   *                                             *
                     :   *         THE A REG CONTAINS THE MUX ADDR     *
                     :   *            IN BITS 7-4  (X0H)               *
                     :   *                                             *
                     :   *                                             *
                     :   ***********************************************
                     :
                     :
01DF    0650     GZ:      MVI     B,50H      :PUT DESIRED MUX ADDR IN B
01F1    B8                CMP     B          :IS THE MUX ADDR 5 ?
01F2    C29302            JNZ     ERROR      :IF NOT, THERE IS AN ERROR
01F5    CD9002            CALL    STRTAD     :START THE NEXT CONVERSION
01F8    212802            LXI     H,ABSPR    :GET ADDR OF "ABS PRESS"
                                             :   SERVICE ROUTINE
01F9    22113C            SHLD    NXTPR      :STORE ADDR IN NXTPR
01EE    2A203C            LHLD    ZG         :LOADS H&L WITH Z-G'S
                                             :   RUNNING SUM
01F1    7B                MOV     A,E        :MOVES CURRENT READING TO A
01F2    CDC902            CALL    SUM        :ADD DATA TO RUNNING SUM
01F5    3A223C            LDA     ZGCNT      :GET Z-G COUNT
01F8    3C                INR     A          :INCREMENT THE COUNT
01F9    FE08              CPI     8
01F9    CA0702            JZ      GZ1        :JUMP IF ZGCNT .EQ. 8
01FE    22203C            SHLD    ZG         :STORE RUNNING SUM IN ZG
0201    32223C            STA     ZGCNT      :STORE COUNT IN ZGCNT
0204    C39C00            JMP     EOC
0207    CDCE02   GZ1:     CALL    AVG        :AVERAGE THE 8 READINGS
020A    AF                XRA     A
020B    32223C            STA     ZGCNT      :ZERO THE COUNT AND
020E    32203C            STA     ZG         :   THE RUNNING SUM
0211    32213C            STA     ZG+1
0214    7B                MOV     A,E        :MOVES AVG DATA TO A
0215    21003C            LXI     H,GZMAX    :GET ADDR OF MAX Z-G'S
0218    BE                CMP     M          :COMPARE AVG VALUE WITH MAX
0219    DA1D02            JC      GZ2        :JUMP IF A.LT.GZMAX
021C    77                MOV     M,A        :REPLACE IF A.GE.GZMAX
021D    21003C   GZ2:     LXI     H,GZMIN    :GET ADDR OF MIN Z-G'S
0220    BE                CMP     M          :COMPARE AVG VALUE WITH M
0221    D29C00            JNC     EOC        :JUMP IF A.GE.GZMIN
0224    77                MOV     M,A        :REPLACE IF A.LT.GZMIN
0225    C39C00            JMP     EOC
```

```
                        ;    ***********************************************
                        ;    *                                             *
                        ;    *                                             *
                        ;    *          THIS IS THE SERVICE ROUTINE       *
                        ;    *             FOR ABS PRESS                   *
                        ;    *                                             *
                        ;    *                                             *
                        ;    *          THIS IS THE LAST CONVERSION       *
                        ;    *             SO NO NEW CONVERSION IS         *
                        ;    *             STARTED                         *
                        ;    *                                             *
                        ;    *          THE CURRENT READING REPLACES      *
                        ;    *             THE OLD.  THE STORAGE           *
                        ;    *             LOCATION IS CABPR               *
                        ;    *                                             *
                        ;    *          THE A REG CONTAINS THE MUX ADDR   *
                        ;    *             IN BITS 7-4  (X0H)              *
                        ;    *                                             *
                        ;    *                                             *
                        ;    ***********************************************
                        ;
                        ;
                        ;
0228    0660            ABSPR:  MVI     B,60H       ;PUT DESIRED MUX ADDR IN B
022A    B8                      CMP     B           ;IS THE MUX ADDR 6 ?
022B    C29302                  JNZ     ERROR       ;IF NOT THERE IS AN ERROR
022E    210E3C                  LXI     H,CABPR     ;GET ADDR OF CURRENT ABS PR
0231    73                      MOV     M,E         ;STORE NEW VALUE IN CABPR
```

```
                          ;   ********************************************
                          ;   *                                          *
                          ;   *                                          *
                          ;   *         THIS IS THE SERVICE ROUTINE      *
                          ;   *            FOR HEART RATE                *
                          ;   *         THIS ROUTINE IS ENTERED RIGHT    *
                          ;   *            AFTER ABS PRESS               *
                          ;   *                                          *
                          ;   *         IF A NEW HEART RATE HAS BEEN     *
                          ;   *            COMPUTED (IBFB=1) IT IS       *
                          ;   *            READ IN.  EIGHT READINGS      *
                          ;   *            ARE AVERAGED TO GIVE AN       *
                          ;   *            AVERAGE HEART RATE.           *
                          ;   *         READINGS .LT. 10H ARE ASSUMED    *
                          ;   *            AS FALSE R-WAVES AND ARE      *
                          ;   *            ADDED TO THE COUNT TO BE      *
                          ;   *            AVERAGED - BUT DO NOT         *
                          ;   *            COUNT AS ONE OF THE 8.        *
                          ;   *         THE NEW AVERAGE HEART RATE       *
                          ;   *            IS THEN STORED.               *
                          ;   *                                          *
                          ;   *                                          *
                          ;   ********************************************
                          ;
                          ;
0232    DBE9      HEART:  IN      STAT2      ;INPUT STATUS OF 8255 #2
0234    E620              ANI     ABFMSK     ;MASKS IBFA
0236    CA6A02            JZ      CNTCK      ;JUMP IF NO NEW HEART RATE
0239    DBE8              IN      HRIN       ;INPUT THE NEW HEART RATE
023B    2F                CMA                ;COMPLEMENTS DATA FROM
                                             ;   INVERTING DRIVER (8226)
023C    2A143C            LHLD    HRT        ;LOADS H&L WITH HEART RATE
                                             ;   RUNNING SUM
023F    CDC302            CALL    SUM        ;ADD DATA TO RUNNING SUM
0242    FE10              CPI     10H        ;(A HAS THE NEW READING)
0244    DA6A02            JC      CNTCK      ;DONE IF A .LT. 10H
0247    3A153C            LDA     HRCNT      ;GET HEART RATE COUNT
024A    3C                INR     A          ;INCREMENT THE COUNT
024B    FE08              CPI     8
024D    CA5902            JZ      HRT1       ;JUMP IF HRCNT .EQ. 8
0250    22143C            SHLD    HRT        ;STORE RUNNING SUM IN HRT
0253    32153C            STA     HRCNT      ;STORE COUNT IN HRCNT
0256    C36A02            JMP     CNTCK
0259    CDCE02    HRT1:   CALL    AVG        ;AVERAGE THE 8 READINGS
025C    210F3C            LXI     H,HRTRT    ;GET ADDR OF CURRENT
                                             ;   HEART RATE
025F    73                MOV     M,E        ;STORE NEW VALUE IN HRTRT
0260    AF                XRA     A
0261    32153C            STA     HRCNT      ;ZERO THE COUNT AND
0264    32143C            STA     HRT        ;   THE RUNNING SUM
0267    32153C            STA     HRT+1
```

```
                     ;   **************************************************
                     ;   *                                                *
                     ;   *                                                *
                     ;   *                                                *
                     ;   *            THIS SECTION DECREMENTS THE         *
                     ;   *               10 SEC COUNTER                   *
                     ;   *                                                *
                     ;   *            EACH 10 SEC THE STORAGE ROUTINE     *
                     ;   *               IS CALLED AND THE COUNTER        *
                     ;   *               IS RELOADED                      *
                     ;   *                                                *
                     ;   *            AFTER COMPLETION OF THIS CODE,      *
                     ;   *               THE PROGRAM HALTS AND WAITS      *
                     ;   *               FOR THE 50 MS TIMER TO EXPIRE    *
                     ;   *                                                *
                     ;   *                                                *
                     ;   **************************************************
                     ;
                     ;
026A    21133C   CNTCK:  LXI    H,TENSC    ;GET ADDR OF 10 SEC COUNT
026D    35               DCR    M          ;DECREMENT THE COUNT
026E    CA7402           JZ     STORE      ;JUMP IF 10 SEC HAS ELAPSED
0271    FB               EI
0272    76               HLT               ;WAIT FOR 50 MS TIMER
0273    00               NOP
```

```
                        :   ***************************************************
                        :   *                                                 *
                        :   *                                                 *
                        :   *            THIS SECTION OUTPUTS THE STORED      *
                        :   *               INFORMATION EVERY 10 SECONDS.     *
                        :   *                                                 *
                        :   *            THE DATA IS OUTPUT IN THE            *
                        :   *               FOLLOWING ORDER:                  *
                        :   *                                                 *
                        :   *               TIMING PREAMBLE (LSB FIRST)       *
                        :   *               PINSM   (LSB FIRST)               *
                        :   *               POUTS   (LSB FIRST)               *
                        :   *               FLRTS   (LSB FIRST)               *
                        :   *               GXMAX   GXMIN   GYMAX   GYMIN     *
                        :   *               GZMAX   GZMIN   ABSPR   HRTRT     *
                        :   *               BTHRT                             *
                        :   *                                                 *
                        :   *                                                 *
                        :   ***************************************************
                        :
                        :
0274    36C8      STORE:  MVI     M,200D      :SETS THE COUNTER TO 200
0276    2A003C            LHLD    TIME        :PLACE OUTPUT COUNT IN H&L
0279    23                INX     H           :INCREMENT THE COUNT
027A    22003C            SHLD    TIME        :STORE THE NEW OUTPUT COUNT
027D    21003C            LXI     H,TIME      :GET ADDR OF FIRST BYTE
0280    1611              MVI     D,BYTES     :LOAD D WITH NUMBER OF
                                              :   OUTPUT BYTES
0282    7E        STOR1:  MOV     A,M         :MOVE THE DATA TO A
0283    CDFE02            CALL    DTOUT       ;OUTPUTS DATA TO CONSOLE
0286    23                INX     H           :GET NEXT OUTPUT BYTE ADDR
0287    15                DCR     D           :DECREMENT THE BYTE COUNTER
0288    C28202            JNZ     STOR1       :JUMP IF MORE OUTPUT
028B    0E7E              MVI     C,CSP
028D    CD1B03            CALL    PRINT       :CAUSES THE DATA ON SCREEN
0290    0E1E              MVI     C,PR        :   TO BE WRITTEN ON TAPE
0292    CD1B03            CALL    PRINT
0295    CDE702            CALL    INIT        :INITIALIZES PINSM TO GZMIN
0298    FB                EI
0299    76                HLT                 :WAIT FOR 50 MS TIMER
029A    00                NOP
```

```
                        ;   *******************************************
                        ;   *                                         *
                        ;   *                                         *
                        ;   *         THE FOLLOWING ERROR ROUTINE IS  *
                        ;   *            EXECUTED WHENEVER THE ACTUAL *
                        ;   *            MUX ADDR IS NOT THE SAME AS  *
                        ;   *            THE EXPECTED MUX ADDR        *
                        ;   *                                         *
                        ;   *         THE A REG CONTAINS THE MUX ADDR IN *
                        ;   *            BITS 7-4 (X0H) AND THE B REG *
                        ;   *            CONTAINS THE DESIRED ADDR    *
                        ;   *         THE ERROR ROUTINE CONVERTS UNTIL *
                        ;   *            THE ACTUAL MUX ADDR EQUALS   *
                        ;   *            THE DESIRED MUX ADDR         *
                        ;   *         THEN THE PROGRAM LOOP IS ENTERED *
                        ;   *            AGAIN VIA THE EOC ROUTINE    *
                        ;   *                                         *
                        ;   *         NOTE:  NXTPR  STILL CONTAINS THE *
                        ;   *            ADDR OF THE DESIRED SERVICE  *
                        ;   *            ROUTINE                      *
                        ;   *                                         *
                        ;   *                                         *
                        ;   *******************************************
                        ;
                        ;
029B    CD8302          ERROR:  CALL    STRTAD      ;STROBE AND CONVERT
029E    DBE5            ERR1:   IN      STAT1       ;INPUT STATUS OF 8255 #1
02A0    E602                    ANI     BBFMSK      ;MASKS IBFB
02A2    CA9E02                  JZ      ERR1        ;LOOP IF STILL CONVERTING
02A5    DBE5                    IN      MUX         ;READS IN MUX ADDR & 4 LSB
02A7    E6F0                    ANI     MUXMSK      ;SAVES B7-B4
02A9    B8                      CMP     B           ;COMPARE NEW ADDR WITH B
02AA    C29B02                  JNZ     ERROR       ;TRY AGAIN IF NOT EQ.
02AD    C39C00                  JMP     EOC         ;REENTER PROGRAM LOOP WHEN
                                                    ;    DESIRED ADDR IS FOUND
```

```
:   **************************************************
:   *                                                *
:   *                                                *
:   *          SYSTEM SUBROUTINES                    *
:   *                                                *
:   *                                                *
:   **************************************************
:   *                                                *
:   *                                                *
:   *        THESE SUBROUTINES ARE USED BY THE       *
:   *   OPERATING SYSTEM AND THE SERVICE ROUTINES    *
:   *   TO ACCOMPLISH TASKS COMMON TO MORE THAN      *
:   *   ONE MODULE.  THERE ARE SUBROUTINES TO        *
:   *   CONTROL THE DAS 1128 (START A CONVERSION     *
:   *   AND INCREMENT THE MUX ADDRESS), MANIP-       *
:   *   ULATE THE DATA (SUMMING AND AVERAGING)       *
:   *   INITIALIZE THE STORAGE AREA, AND OUTPUT      *
:   *   THE DATA (FORMATTING THE DATA AND            *
:   *   SENDING IT TO THE CONSOLE                    *
:   *                                                *
:   *                                                *
:   **************************************************
```

```
                    ;    ************************************************
                    ;    *                                              *
                    ;    *                                              *
                    ;    *         THIS SUBROUTINE INCREMENTS THE       *
                    ;    *            DAS 1128 MUX ADDR AND STARTS      *
                    ;    *            THE CONVERSION                    *
                    ;    *                                              *
                    ;    *                                              *
                    ;    ************************************************
                    ;
                    ;
0290    CD9702      STRTAD: CALL     STROBE     ;INCREMENT THE MUX ADDR
0293    CDC002              CALL     TRIGGR     ;STARTS THE CONVERSION
0296    C9                  RET
                    ;
                    ;
                    ;    ************************************************
                    ;    *                                              *
                    ;    *                                              *
                    ;    *         THIS SUBROUTINE INCREMENTS THE       *
                    ;    *            MUX ADDR OF THE DAS 1128          *
                    ;    *                                              *
                    ;    *                                              *
                    ;    ************************************************
                    ;
                    ;
0297    3E0D        STROBE: MVI      A,STBON    ;SETS BIT C6 WHICH INCRE-
0299    D3E7                OUT      ADCON      ;   MENTS MUX ADDR BY 1
029B    3E0C                MVI      A,STBOFF   ;RESETS BIT C5
029D    D3E7                OUT      ADCON
029F    C9                  RET
                    ;
                    ;
                    ;    ************************************************
                    ;    *                                              *
                    ;    *                                              *
                    ;    *         THIS SUBROUTINE TRIGGERS THE DAS     *
                    ;    *            1128 WHICH STARTS A CONVERSION    *
                    ;    *                                              *
                    ;    *                                              *
                    ;    ************************************************
                    ;
                    ;
02C0    3E0F        TRIGGR: MVI      A,TRGON    ;TURNS ON BIT C7 (TRIGGER)
02C2    D3E7                OUT      ADCON      ;   TO START A CONVERSION
02C4    3E0E                MVI      A,TRGOFF   ;TURNS OFF BIT C7
02C6    D3E7                OUT      ADCON
02C8    C9                  RET
```

```
                  ;    ***********************************************
                  ;    *                                             *
                  ;    *                                             *
                  ;    *           THIS SUBROUTINE TAKES INPUT DATA  *
                  ;    *              FROM REG A AND ADDS IT TO H&L  *
                  ;    *                                             *
                  ;    *                                             *
                  ;    ***********************************************
                  ;
                  ;
02C9    4F        SUM:     MOV     C,A          ;MOV DATA TO C
02CA    0600               MVI     B,0          ;ZERO OUT B
02CC    09                 DAD     B            ;H&L = H&L + B&C
02CD    C9                 RET
                  ;
                  ;
                  ;    ***********************************************
                  ;    *                                             *
                  ;    *                                             *
                  ;    *           THIS SUBROUTINE TAKES THE RUNNING *
                  ;    *              SUM LOCATED IN H&L AND DIVIDES *
                  ;    *              IT BY 8.  THE RESULTING AVERAGE *
                  ;    *              IS RETURNED TO THE CALLING     *
                  ;    *              PROGRAM IN REG E.              *
                  ;    *                                             *
                  ;    *                                             *
                  ;    ***********************************************
                  ;
                  ;
02CE    7D        AVG:     MOV     A,L          ;OBTAIN LSB OF SUM
02CF    0F                 RRC
02D0    0F                 RRC                  ;DIVIDE BY 8
02D1    0F                 RRC
02D2    D2DD02             JNC     AVG1         ;JUMP IF REMAINDER .LT. 4
02D5    F6E0               ORI     0E0H         ;INSURE THAT INCR WILL OVER
02D7    C601               ADI     1            ;   FLOW IF 5 LSB'S ARE 1'S
02D9    D2DD02             JNC     AVG1
02DC    24                 INR     H            ;ADD "CARRY" TO MSB OF SUM
02DD    E61F      AVG1:    ANI     1FH          ;DISCARD REMAINDER
02DF    5F                 MOV     E,A          ;STORE LSB IN E
02E0    7C                 MOV     A,H          ;OBTAIN MSB OF SUM
02E1    0F                 RRC
02E2    0F                 RRC                  ;ROTATE TO MSB OF A
02E3    0F                 RRC
02E4    83                 ADD     E            ;ADD LSB TO MSB
02E5    5F                 MOV     E,A          ;MOVES AVG READING TO E
02E6    C9                 RET
```

```
                        ;   ***************************************************
                        ;   *                                                 *
                        ;   *                                                 *
                        ;   *         THIS SECTION ZEROES THE FOLLOW-         *
                        ;   *            ING LOCATIONS:  PINSM, POUTS         *
                        ;   *            FLRTS, GXMAX, GYMAX, & GZMAX         *
                        ;   *         0FFH IS PLACED IN THE FOLLOWING         *
                        ;   *            LOCATIONS:  GXMIN, GYMIN,            *
                        ;   *            GZMIN                                *
                        ;   *                                                 *
                        ;   *                                                 *
                        ;   ***************************************************
                        ;
                        ;
                        ;
02E7    AF              INIT:   XRA     A
02E8    210230                  LXI     H,PINSM     ;GET ADDR OF START OF
                                                    ;   STORAGE TO BE CLEARED
02EB    1603                    MVI     D,LOCS      ;# OF LOCATIONS
02ED    77              INIT1:  MOV     M,A
02EE    23                      INX     H           ;ZEROES OUT THE STORAGE
02EF    15                      DCR     D           ;   AREA
02F0    C2ED02                  JNZ     INIT1
02F3    16FF                    MVI     D,0FFH      ;LOADS D WITH FF
02F5    72                      MOV     M,D         ;MOVES FF TO GZMIN
02F6    210930                  LXI     H,GXMIN
02F9    72                      MOV     M,D         ;MOVES FF TO GXMIN
02FA    23                      INX     H
02FB    23                      INX     H
02FC    72                      MOV     M,D         ;MOVES FF TO GYMIN
02FD    C9                      RET
                        ;
                        ;
                        ;   ***************************************************
                        ;   *                                                 *
                        ;   *                                                 *
                        ;   *                                                 *
                        ;   *         THIS SUBROUTINE SEPARATES THE           *
                        ;   *            8 BITS OF DATA IN THE A REG          *
                        ;   *            INTO TWO 4 BIT NUMBERS -             *
                        ;   *            EACH OF THESE NUMBERS IS             *
                        ;   *            CONVERTED TO AN ASCII                *
                        ;   *            CHARACTER AND OUTPUTED TO            *
                        ;   *            THE CONSOLE.                         *
                        ;   *                                                 *
                        ;   *                                                 *
                        ;   *         (BORROWED FROM THE SBC                  *
                        ;   *          80P20 USERS GUIDE                      *
                        ;   *          PAGE 8-29 (REF 15))                    *
                        ;   *                                                 *
                        ;   *                                                 *
                        ;   ***************************************************
                        ;
                        ;
02FE    F5              OTOUT:  PUSH    PSW         ;SAVE ARGUMENT
02FF    0F                      RRC
0300    0F                      RRC                 ;MOVES MSB'S TO LOW 4 BITS
0301    0F                      RRC
0302    0F                      RRC
0303    CD1103                  CALL    ASCII       ;CONVERT 4 LSB'S TO ASCII
0306    CD1B03                  CALL    PRINT       ;OUTPUT TO TERMINAL
0309    F1                      POP     PSW         ;GET BACK ARGUMENT
030A    CD1103                  CALL    ASCII       ;CONVERT 4 LSB'S TO ASCII
030D    CD1B03                  CALL    PRINT       ;OUTPUT TO TERMINAL
0310    C9                      RET
```

```
                        ;      ***********************************************
                        ;      *                                             *
                        ;      *                                             *
                        ;      *            THIS SUBROUTINE CONVERTS THE 4   *
                        ;      *               LSB'S OF THE A REG (0-9,A-F)  *
                        ;      *               INTO THE CORRESPONDING ASCII  *
                        ;      *               CHARACTER (30-39,41-46)       *
                        ;      *                                             *
                        ;      *            THE DAA (DECIMAL ADJUST ACC)     *
                        ;      *               INSTRUCTION PERFORMS THE      *
                        ;      *               FOLLOWING:                    *
                        ;      *                  1. IF THE VALUE OF THE     *
                        ;      *                     4 LSB'S OF THE ACC      *
                        ;      *                     IS .GT. 9 OR IF THE     *
                        ;      *                     'AC' FLAG IS SET, 6     *
                        ;      *                     IS ADDED TO THE ACC     *
                        ;      *                  2. IF THE VALUE OF THE     *
                        ;      *                     4 MSB'S OF THE ACC      *
                        ;      *                     IS NOW .GT. 9, OR       *
                        ;      *                     IF THE 'CY' FLAG IS     *
                        ;      *                     SET, 6 IS ADDED TO      *
                        ;      *                     THE 4 MSB'S OF ACC      *
                        ;      *                                             *
                        ;      *                                             *
                        ;      *            (BORROWED FROM THE SBC           *
                        ;      *             80P20 USERS GUIDE               *
                        ;      *             PAGE B-33 (REF 15))             *
                        ;      *                                             *
                        ;      *                                             *
                        ;      ***********************************************
                        ;
                        ;
0311    E60F            ASCII:  ANI     0FH         ;SAVES 4 LSB'S (1 HEX CHAR)
0313    C690                    ADI     90H         ;INSURE THAT A-F CAUSES
                                                    ;   A CARRY
0315    27                      DAA                 ;ADJUST CONTENTS OF A REG
0316    CE40                    ACI     40H         ;ADD IN CARRY AND ADJUST
                                                    ;   THE UPPER 4 BITS
0318    27                      DAA                 ;ADJUST CONTENTS OF A REG
0319    4F                      MOV     C,A         ;MOVE ASCII CHAR TO C
031A    C9                      RET
                        ;
                        ;
                        ;      ***********************************************
                        ;      *                                             *
                        ;      *                                             *
                        ;      *            THIS SUBROUTINE WAITS UNTIL THE  *
                        ;      *               CONSOLE IS READY TO ACCEPT A  *
                        ;      *               CHARACTER AND THEN OUTPUTS THE*
                        ;      *               CHARACTER (STORED IN C REG)   *
                        ;      *               TO THE CONSOLE                *
                        ;      *                                             *
                        ;      *                                             *
                        ;      ***********************************************
                        ;
                        ;
031B    DBED            PRINT:  IN      USART       ;INPUT STATUS OF CONSOLE
031D    E601                    ANI     READY       ;MASK FOR TRANSMITTER READY
031F    CA1B03                  JZ      PRINT       ;LOOP IF NOT READY
0322    79                      MOV     A,C         ;MOVE CHARACTER TO A REG
0323    D3EC                    OUT     CON         ;SEND TO CONSOLE
0325    C9                      RET
```

```
                          ;  ****************************************
                          ;  *                                      *
                          ;  *                                      *
                          ;  *     THIS SECTION CONTAINS THE        *
                          ;  *        JUMP TABLE FOR THE            *
                          ;  *       INTERRUPT CONTROLLER           *
                          ;  *                                      *
                          ;  *                                      *
                          ;  ****************************************
                          ;
                          ;
03E0                         ORG     03E0H
03E0    C30000               JMP     0          ;TEMP FOR LEVEL 0
03E3    00                   NOP
03E4    C30000               JMP     0          ;TEMP FOR LEVEL 1
03E7    00                   NOP
03E8    C30000               JMP     0          ;TEMP FOR LEVEL 2
03EB    00                   NOP
03EC    C37E00               JMP     T50MS      ;SERVICE ROUT FOR LEVEL 3
03EF    00                   NOP
03F0    C30000               JMP     0          ;TEMP FOR LEVEL 4
03F3    00                   NOP
03F4    C30000               JMP     0          ;TEMP FOR LEVEL 5
03F7    00                   NOP
03F8    C30000               JMP     0          ;TEMP FOR LEVEL 6
03FB    00                   NOP
03FC    C30000               JMP     0          ;TEMP FOR LEVEL 7
03FF    00                   NOP
```

```
                             ;   *********************************************
                             ;   *                                           *
                             ;   *                                           *
                             ;   *         THIS SECTION CONTAINS THE STORAGE *
                             ;   *            AREA FOR THE PROGRAM (CPU      *
                             ;   *            SCRATCH PAD STORAGE)           *
                             ;   *                                           *
                             ;   *                                           *
                             ;   *********************************************
                             ;   *                                           *
                             ;   *                                           *
                             ;   *         THESE ARE THE LOCATIONS THAT      *
                             ;   *            WILL BE STORED EVERY 10 SEC    *
                             ;   *                                           *
                             ;   *                                           *
                             ;   *********************************************
                             ;
                             ;
3C00                                 ORG     3C00H
3C00      0000               TIME:   DW      0               ;CONTAINS RUNNING OUTPUT
                                                             ;   COUNT.  EACH COUNT
                                                             ;   REPRESENTS 10 SEC
3C02      0000               PINSM:  DW      0               ;CONTAINS PO2IN RUNNING SUM
3C04      0000               POUTS:  DW      0               ;CONTAINS PO2OUT RUNNING SM
3C06      0000               FLRTS:  DW      0               ;CONTAINS FLOW RATE RUNNING
                                                             ;   SUM
3C08      00                 GXMAX:  DB      0               ;CONTAINS MAX X-G'S
3C09      00                 GXMIN:  DB      0               ;CONTAINS MIN X-G'S
3C0A      00                 GYMAX:  DB      0               ;CONTAINS MAX Y-G'S
3C0B      00                 GYMIN:  DB      0               ;CONTAINS MIN Y-G'S
3C0C      00                 GZMAX:  DB      0               ;CONTAINS MAX Z-G'S
3C0D      00                 GZMIN:  DB      0               ;CONTAINS MIN Z-G'S
3C0E      00                 CABPR:  DB      0               ;CONTAINS CURRENT ABS PRESS
3C0F      00                 HRTRT:  DB      0               ;CONTAINS CURRENT HEART
                                                             ;   RATE
3C10      00                 BTHRT:  DB      0               ;CONTAINS BREATH RATE
```

```
                          ;   **********************************************
                          ;   *                                            *
                          ;   *                                            *
                          ;   *         THIS IS ADDITIONAL STORAGE         *
                          ;   *                                            *
                          ;   *                                            *
                          ;   **********************************************
                          ;
                          ;
3C11    0000      NXTPR:  DW      0          ;CONTAINS EOC SERVICE
                                             ;   ROUTINE ADDRESS
3C13    00        TENSC:  DB      0          ;CONTAINS 10 SEC COUNT
3C14    0000      HRT:    DW      0          ;CONTAINS HEART RATE
                                             ;   RUNNING SUM
3C16    00        HRCNT:  DB      0          ;CONTAINS HEART RATE
                                             ;   COUNT (0-8)
3C17    00        TMCNT:  DB      0          ;CONTAINS 50 MS COUNT
3C18    00        THCNT:  DB      0          ;CONTAINS THRESHOLD CNT
3C19    00        TFLAG:  DB      0          ;CONTAINS THRESHOLD FLAG
3C1A    0000      XG:     DW      0          ;CONTAINS X-G'S RUNNING SUM
3C1C    00        XGCNT:  DB      0          ;CONTAINS X-G'S COUNT (0-8)
3C1D    0000      YG:     DW      0          ;CONTAINS Y-G'S RUNNING SUM
3C1F    00        YGCNT:  DB      0          ;CONTAINS Y-G'S COUNT (0-8)
3C20    0000      ZG:     DW      0          ;CONTAINS Z-G'S RUNNING SUM
3C22    00        ZGCNT:  DB      0          ;CONTAINS Z-G'S COUNT (0-8)
                          END
NO PROGRAM ERRORS
```

```
                              SYMBOL TABLE

* 01

A       0007      ABFMS   0020      ABSPR   0228      ADCON   00E7
AIBI    00B5      AIBO    00B4      ASCII   0311      AVG     02CE
AVG1    0200      B       0000      BBFMS   0002      BRRT    0111 *
BRRT1   0125      BTHRT   3C10      BYTES   0011      C       0001
C0M3    0036      C1M3    0070      C2M3    0036      C6OFF   000C
C7OFF   000E      CABPR   3C0E      CNTCK   025A      CON     00EC
CSP     007E      CTRS    000C      CTR1    000D      CTR2    000E
D       0002      DATA    00E4      DTOUT   02FE      E       0003
EOC     009C      EOC1    00A1      EOC2    0038      EOIC    0020
ERR1    029E      ERROR   0298      FLAG    00FF      FLRTS   3C06
FLHRT   00FB      GX      0140      GX1     0175      GX2     0188
GXMAX   3C08      GXMIN   3C09      GY      0196      GY1     01BE
GY2     01D4      GYMAX   3C0A      GYMIN   3C0B      GZ      01DF
GZ1     0207      GZ2     021D      GZMAX   3C0C      GZMIN   3C0D
H       0004      HEART   0232 *    HRCNT   3C16      HRIN    00E8
HRT     3C14      HRT1    0259      HRTRT   3C0F      ICCP1   00DA
ICCP2   00DB      ICW1    00F6      ICW2    0003      IDA     0059
IDA1    0C6D      IMASK   00C4      INIT    02E7      INIT1   02ED
L       0005      LOCS    000B      M       0006      MODE    004E
MSKPT   00D9      MUX     00E5      MUXMS   00F0      NXTPR   3C11
PINSM   3C02      POZIN   00C9      POZOU   00E2      POUTS   3C04
PPI1    00E7      PPI2    00EB      PR      001E      PRBCN   0060
PRINT   031B      PSW     0006      PWRUP   0045      READY   0001
RSTUP   0037      SP      0006      STACK   3F80      STAT1   00E6
STAT2   00EA      STBOF   00CC      STBON   000D      STOR1   0282
STORF   0274      STROB   02B7      STRTA   0290      SUM     02C9
T50MS   007E      TENSC   3C13      TFLAG   3C19      THCNT   3C18
TIME    3C00      TMCNT   3C17      TMCP    000F      TRGOF   000E
TRGON   000F      TRHLD   0020      TRIGG   0200      USART   00ED
XG      3C1A      XGCNT   3C1C      YG      3C1D      YGCNT   3C1F
ZG      3C20      ZGCNT   3C22
```

Appendix C

IFPDAS II Prototype User's Guide

Data Collection

This section describes the step-by-step procedure necessary to operate the IFPDAS II prototype in its data acquisition function.

1. AC Power - ON to both the Hazeltine video terminal and
   the cassette tape

2. DC Power - +5 VDC supply: OFF
   all others (-5, +12, -12, +15, & -15 VDC): ON

3. Video Terminal - Parity: 1
   Full Duplex
   Baud Rate: 1200
   Clear the screen

4. Cassette Recorder - Insert tape in Recorder 2 and engage
   Select CONT - OFF LINE - PAGE
   Depress RESET, then REWIND
   Depress INTERLOCK and RECORD button
   and wait for tape to stop
   (RECORD button stays lighted)

5. +5 VDC Supply - ON

The +5 VDC power supply resets the SBC 80/20 hardware and the operating system starts to execute. Data will be output to the video terminal every ten seconds and transferred from the screen to the cassette tape under program control. The 8080 CPU directs the writing of this data using two control characters. The CONTROL SHIFT PERIOD (cs.) character (7EH) tells the Hazeltine that a command will follow. The PRINT character (1EH) directs the transfer from the screen to the tape.

When all of the desired data has been recorded:

1. DC Power - OFF (this terminates the IFPDAS's operation)
2. Push RESET (the tape will advance momentarily, then stop and the RECORD light will go out)
3. Press REWIND
4. AC Power - OFF (to both the video terminal and the cassette recorder)

The data is now recorded on the cassette tape and is ready for transfer to the main computer.

Data Transfer

This section describes the step-by-step procedure to transfer the data from the cassette tape to a permanent file on AFIT's computer system.

1. Video Terminal - Parity: 1
   Half Duplex
   Baud Rate: 300
2. Cassette Recorder - Select CONT - OFF LINE - PAGE
   Depress RESET, then REWIND
3. Using the terminal, LOGIN and enter EDITOR
4. Enter: CREATE, SUPRESS line numbers (C,S)
5. After system responds ENTER LINES, depress PLAYBACK on tape channel
6. After the data is transferred, send an "=" to release the CREATE mode
7. List the file and check for errors

The data is now in the edit file. To store it permanently, enter the following commands:

1. REQUEST,Q,*PF

```
2. SAVE,Q,NOSEQ,O       (NOSEQuence, Overwrite)

3. CATALOG,Q,DATA,ID=(problem #),RP=(# of days to retain)
```

The data is now stored on disc for later use by the post-flight data conversion routine.

Data Conversion

This section describes the procedure to execute the compiled post-flight data conversion routine (COMPCONVERT), using the file DATA as the data.

Enter the following commands:

```
1. ATTACH,LGO,COMPCONVERT        (attaches COMPCONVERT as a local
                                  file called LGO)

2. ATTACH,AFITSUBROUTINES,ID=AFIT        (attaches the AFIT sub-
                                          routines as a local file
                                          called AFITSUB)

3. LIBRARY,AFITSUB

4. ATTACH,TAPE10,DATA         (attaches DATA as a local file called
                               TAPE10)

5. REWIND,LGO

6. REWIND,TAPE10

7. LGO                        (executes the post-flight data
                               conversion routine)
```

When the program completes its execution, it will have created a local file called PLOT containing the output graphs. To send these graphs to AFIT's plotter, enter:

```
ROUTE,PLOT,TID=BB,FID=(xxx),DC=PT
```

This routing completes the data conversion process.

Appendix D

Post Flight Data Conversion Routine

This program takes the data from the cassette tape (TAPE10) and performs the post flight data conversions. In addition to the calculations, each program module formats the labels for the graphs and calls the plotting routines.

The following parameters are plotted versus time by this program:

Cabin Absolute Pressure
Cabin Altitude
Z - G's
Heart Rate
Breathing Rate
Minute Ventilation Volume
Inspired Oxygen Quantity
Expired Oxygen Quantity

A flow chart of the post flight data conversion routine is given in Figure 20; a listing of the program follows the flow chart.

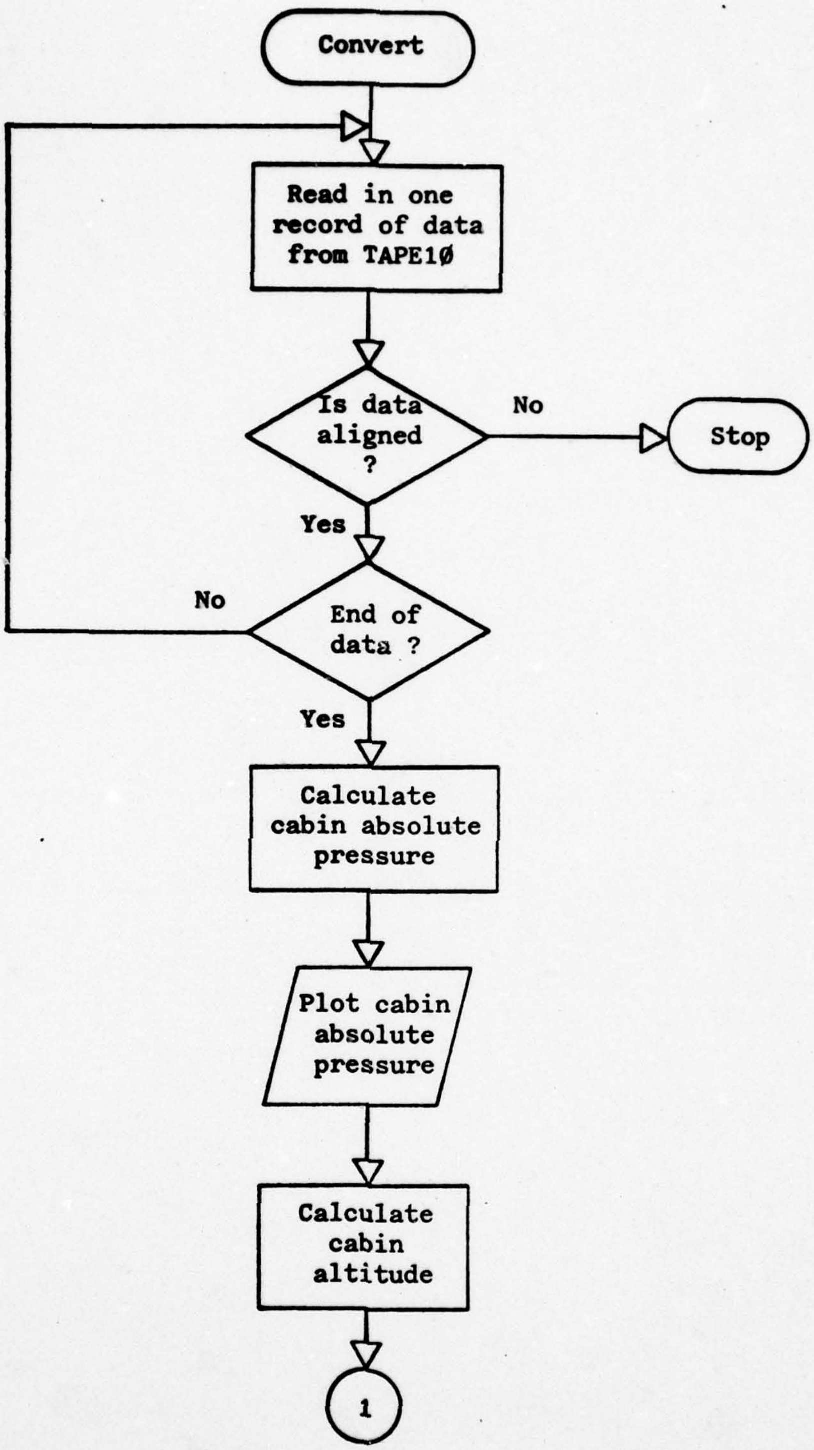


Fig. 20. Post Flight Data Conversion Flow Chart (Sheet 1 of 2)

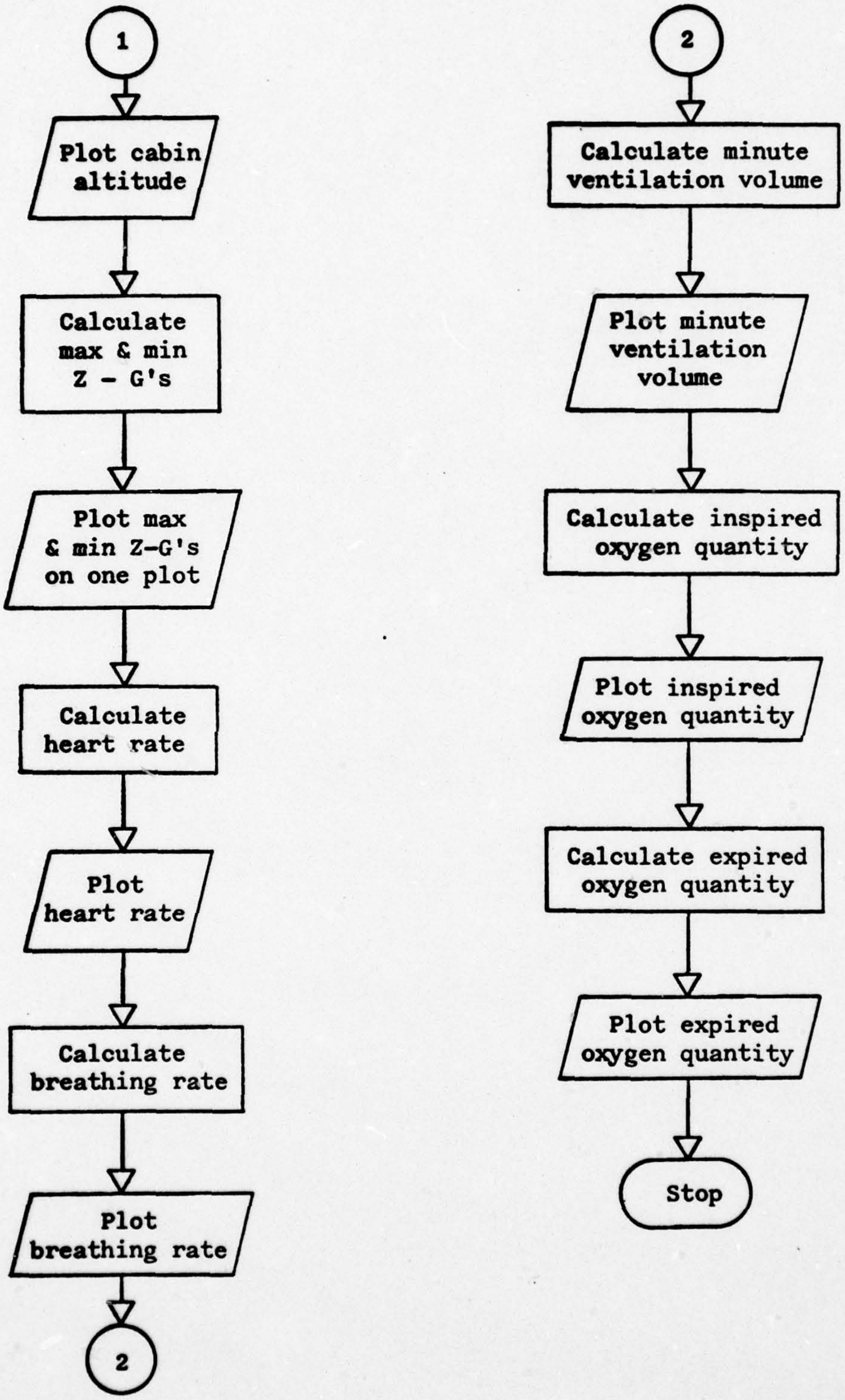


Fig. 20. Post Flight Data Conversion Flow Chart (Sheet 2 of 2)

```
C            POST FLIGHT DATA CONVERSION
C
C
C      THIS PROGRAM TAKES THE DATA FROM THE CASSETTE
C      TAPE (TAPE10) AND PERFORMS THE POST FLIGHT DATA
C      CONVERSIONS.  IN ADDITION TO THE CALCULATIONS,
C      EACH PROGRAM MODULE FORMATS THE LABELS FOR THE
C      GRAPHS AND CALLS THE PLOTTING ROUTINES.
C
C
C      THE FOLLOWING PARAMETERS ARE PLOTTED
C      VERSUS TIME BY THIS PROGRAM:
C
C            CABIN ABSOLUTE PRESSURE
C
C            CABIN ALTITUDE
C
C            Z-G'S
C
C            HEART RATE
C
C            BREATHING RATE
C
C            MINUTE VENTILATION VOLUME
C
C            INSPIRED OXYGEN QUANTITY
C
C            EXPIRED OXYGEN QUANTITY
C
C
      PROGRAM CONVERT(INPUT,OUTPUT,TAPE10)
      INTEGER TIME1,TIME2,A(100,15)
      DIMENSION X(102),Y(102),ID(17),IBUF(100),B(107,2)
      NROW=0
      DO 20 I=1,100
            READ(10,100) TIME1,TIME2,(A(I,J),J=1,15)
100   FORMAT(17Z2)
            IF(EOF(10))30,10
10          K=TIME1+256*TIME2
            IF(K.NE.I) STOP "DATA MISALIGNED"
            NROW=NROW+1
20    CONTINUE
```

```
C              CABIN ABSOLUTE PRESSURE
C
C        THE CABIN ABSOLUTE PRESSURE IS CALCULATED
C        USING THE FOLLOWING FORMULA:
C
C                                          (DATA)
C        CABIN ABSOLUTE PRESSURE = ------ X 760
C                                           250
C
C
30       ID(1)="ABS PRESSU"
         ID(2)="RE VS TIME"
         ID(3)="SUBJECT:  "
         ID(4)="JGJ       "
         ID(5)="START TIME"
         ID(6)=":  1415    "
         ID(7)="     19 OCT"
         ID(8)=" 77       "
         ID(9)="      TIME "
         ID(10)=" (MIN)     "
         ID(11)="ABS PRESSU"
         ID(12)="RE (MM HG)"
         ID(13)="          "
         ID(14)="          "
         ID(15)="          "
         ID(16)="          "
         ID(17)="          "
         DO 60 I=1,NROW
            Y(I)=A(I,13)*760./250.
            B(I,1)=Y(I)
            X(I)=I/6.
60       CONTINUE
         X(NROW+1)=0.
         Y(NROW+1)=250.
         X(NROW+2)=1.
         Y(NROW+2)=100.
         CALL PLOTS(IBUF,100,4HPLOT)
         CALL PLOT (0.,-4.,-3)
         CALL PLOT (0.,0.03,-3)
         CALL HGRAPH (X,Y,NROW,ID,-1,0,0)
```

```
C                   CABIN ALTITUDE
C
C       THE CABIN ABSOLUTE PRESSURE IS CONVERTED TO
C       ALTITUDE (IN FEET) USING THE FOLLOWING
C       RELATIONSHIP:
C
C       ALTITUDE (FEET) = 170,156 - 25,685 LN (ABS PR (MM HG))
C
C
        ID(1)="  ALTITUDE"
        ID(2)=" VS TIME  "
        ID(11)="  ALTITUDE"
        ID(12)="  (FEET)  "
        DO 65 I=1,NROW
              Y(I)=170516-25685*(ALOG(B(I,1)))
              X(I)=I/6.
65      CONTINUE
        X(NROW+1)=0.
        Y(NROW+1)=0.
        X(NROW+2)=1.
        Y(NROW+2)=5000.
        CALL PLOT (0.,-4.,-3)
        CALL PLOT (0.,0.03,-3)
        CALL HGRAPH (X,Y,NROW,ID,-1,0,0)
```

```
C                   Z - G'S
C
C     THE ACCELERATION DATA IS CONVERTED TO G'S USING
C     THE FOLLOWING FORMULA:
C
C                              (DATA)
C     ACCELERATION(G'S) = ( ------ X 15 )   - 3
C                               250
C
C     THE MAXIMUM AND MINIMUM VALUES ARE PLOTTED ON THE
C     SAME GRAPH FOR COMPARISON.
C
C
      ID(1)="   Z-G'S  "
      ID(2)="VS  TIME  "
      ID(11)="MAX AND MI"
      ID(12)="N  Z-G'S  "
      DO 120 I=1,NROW
           Y(I)=(A(I,11)*.06)-3.
           X(I)=I/6.
120   CONTINUE
      X(NROW+1)=0.
      Y(NROW+1)=0.5
      X(NROW+2)=1.
      Y(NROW+2)=.4
      CALL PLOT (0.,-4.,-3)
      CALL PLOT (0.,0.03,-3)
      CALL HGRAPH (X,Y,NROW,ID,-1,0,0)
C
      ID(1)=0
      DO 130 I=1,NROW
           Y(I)=(A(I,12)*.06)-3.
           X(I)=I/6.
130   CONTINUE
      X(NROW+1)=0.
      Y(NROW+1)=0.5
      X(NROW+2)=1.
      Y(NROW+2)=.4
      CALL PLOT (0.,-4.,-3)
      CALL PLOT (0.,0.03,-3)
      CALL HGRAPH (X,Y,NROW,ID,2,1,1)
```

```
C                     HEART RATE
C
C     EACH HEART RATE COUNT REPRESENTS 4.44 MSEC.  THE
C     COUNT IS CONVERTED TO THE HEART RATE USING THE
C     FOLLOWING EQUATION:
C
C                                                  1
C     HEART RATE (BEATS/MIN) = ------------------ X 60
C                              (4.44 MSEC)(COUNT)
C
C
      ID(1)=" HEART RAT"
      ID(2)="E VS TIME "
      ID(11)="HEART RATE"
      ID(12)="  (B/MIN)  "
      DO 40 I=1,NROW
          Y(I)=(225.*60.)/A(I,14)
          X(I)=I/6.
40    CONTINUE
      CALL PLOT(0.,-4.,-3)
      CALL PLOT(0.,0.03,-3)
      CALL HGRAPH(X,Y,NROW,ID,1,0,0)
```

```
C                     BREATHING RATE
C
C     EACH COUNT IS WORTH 50 MSEC.  THE BREATHING RATE
C     IS CALCULATED AS FOLLOWS:
C
C                                                       1
C     BREATHING RATE (BREATHS/MIN) = ---------------- X 60
C                                    (50 MSEC)(COUNT)
C
C
      ID(1)="BREATH RAT"
      ID(11)="BREATH RAT"
      ID(12)="E  (B/MIN)"
      DO 50 I=1,NROW
          Y(I)=(20.*60.)/A(I,15)
          X(I)=I/6.
50    CONTINUE
      CALL PLOT (0.,-4.,-3)
      CALL PLOT (0.,0.03,-3)
      CALL HGRAPH (X,Y,NROW,ID,1,0,0)
```

```
C            MINUTE VENTILATION VOLUME
C
C     THE MINUTE VENTILATION VOLUME IS CALCULATED USING
C     THE IFPDAS-SUPPLIED SUMMATION AND THE ABSOLUTE
C     PRESSURE DATA.  THE FORMULA IS:
C
C                                (SUM)
C     VOLUME    10 SEC (L) = ------------- X .0114
C                            SQRT(ABS PR)
C
C
      DO 70 I=1,NROW
           B(I+3,2)=(A(I,5)+256*A(I,6))/(SQRT(B(I,1))*87.75)
70    CONTINUE
      B(1,2)=B(4,2)
      B(2,2)=B(4,2)
      B(3,2)=B(4,2)
      B(NROW+4,2)=B(NROW+3,2)
      B(NROW+5,2)=B(NROW+3,2)
      DO 80 I=1,NROW
           B(I,2)=B(I,2)+B(I+1,2)+B(I+2,2)+B(I+3,2)+B(I+4,2)
     &                                        +B(I+5,2)
           Y(I)=B(I,2)
           X(I)=I/6.
80    CONTINUE
      ID(1)="MIN VENT V"
      ID(2)="OL VS TIME"
      ID(11)="MIN VENT V"
      ID(12)="OL   (L)   "
      CALL PLOT (0.,-4.,-3)
      CALL PLOT (0.,0.03,-3)
      CALL HGRAPH (X,Y,NROW,ID,1,0,0)
```

```
C               INSPIRED OXYGEN QUANTITY
C
C     THE QUANTITY OF INSPIRED OXYGEN IS COMPUTED BY:
C
C                               (PO2IN SUM)(MIN VENT VOL)
C     QUANTITY OF O2 (L) = ------------------------
C                                 65.79 (ABS PR)
C
C
      ID(1)="OXYGEN INT"
      ID(2)="AKE VS T  "
      ID(11)="OXYGEN INT"
      ID(12)="AKE   (L)  "
      DO 90 I=1,NROW
          Y(I)=((A(I,1)+256*A(I,2))*3.04*B(I,2))
     &                        /(200.*B(I,1))
          X(I)=I/6.
90    CONTINUE
      CALL PLOT (0.,-4.,-3)
      CALL PLOT (0.,0.03,-3)
      CALL HGRAPH (X,Y,NROW,ID,1,0,0)




C                 EXPIRED OXYGEN QUANTITY
C
C     THE QUANTITY OF EXPIRED OXYGEN IS COMPUTED BY:
C
C                               (PO2OUT SUM)(MIN VENT VOL)
C     QUANTITY OF O2 (L) = --------------------------
C                                   65.79 (ABS PR)
C
C
      ID(1)="OXYGEN EXP"
      ID(2)="IRED VS T "
      ID(11)="OXYGEN EXP"
      ID(12)="IRED   (L) "
      DO 110 I=1,NROW
          Y(I)=((A(I,3)+256*A(I,4))*3.04*B(I,2))
     &                        /(200.*B(I,1))
          X(I)=I/6.
110   CONTINUE
      CALL PLOT (0.,-4.,-3)
      CALL PLOT (0.,0.03,-3)
      CALL HGRAPH (X,Y,NROW,ID,1,0,0)
C
      STOP  "YOU MADE IT"
      END
```

The following subroutines are used by the post flight data conversion routine to format and plot the graphs. These subroutines were borrowed from AFIT's EE 6.91 course.

```
      SUBROUTINE HGRAPH(X,Y,N,ID,NO,NP,NS)
      DIMENSION X(1),Y(1),ID(1) $ IF (NO.EQ.2) CALL PLOT(-1.85,2.10,-3)
      IF (NO.EQ.2) GO TO 30       $ IF (NO.LT.0) GO TO 10
      CALL SCALE(X,7.,N,1)        $ CALL SCALE(Y,5.,N,1)
10    CALL PLOT(0.,11.,2)  $  CALL PLOT(8.5,11.,2)
      CALL PLOT(8.5,0.,2) $  CALL PLOT(0.,0.,2)
      CALL PLOT(1.35,1.35,-3)  $  CALL PLOT(0.,8.30,-2)
      IF(ID(1).EQ.000) GO TO 25
      CALL PLOT(.1,-.1,-3)        $ CALL PLOT(0.,-2.,-2)
      DO 20 I=1,7,2
20    CALL SYMBOL( (I+1.5)*.1,.3,.07,ID(I),90.,20)
      CALL PLOT(0.,0.,3)          $ CALL PLOT(1.,0.,2)
      CALL PLOT(1.,2.,2)          $ CALL PLOT(0.,2.,-2)
      CALL PLOT(-.1,.1,-3)
25    CALL PLOT(5.8,0.,-2)
      CALL PLOT(0.,-8.30,-2)  $  CALL PLOT(-5.8,0.,-2)
      CALL SYMBOL(.5,-.2,.1,ID(13),0.,50) $ CALL PLOT(5.3,.75,-3)
      CALL AXIS(0.,0.,ID(9),-20,7.,90.,X(N+1),X(N+2))
      CALL AXIS(0.,0.,ID(11),20,5.,180.,Y(N+1),Y(N+2))
30    Y(N+2)=-Y(N+2)              $ CALL LINE(Y,X,N,1,NP,NS)
      Y(N+2)=-Y(N+2)              $ CALL PLOT(1.85,-2.10,-3)
      RETURN                      $ END
```

```
        SUBROUTINE AXIS(X0,Y0,L,NC,RL,ANG,RMIN,DR)
        DIMENSION L(1) $ A=ANG*3.14159/180. $ DX=.1*COS(A) $ DY=.1*SIN(A)
        IC=ISIGN(1,NC) $ NNC=IABS(NC) $ R=.1 $ N=1 $ X=X0 $ Y=Y0$
10      CALL PLOT(X,Y,3) $ X=X+DX $ Y=Y+DY $ CALL PLOT(X,Y,2)
        CALL PLOT(X-.21*DY*IC,Y+.21*DX*IC,2)
        IF(N.EQ.5) CALL PLOT(X-.42*DY*IC,Y+.42*DX*IC,2)
        IF(N.EQ.10) CALL PLOT(X-.70*DY*IC,Y+.70*DX*IC,2)
        N=MOD(N,10)+1 $ R=R+.1 $ IF(R.LT.RL) GO TO 10
        A=ANG-(IC+1)*45. $ DX=10.*DX $ DY=10.*DY
        C=-.175+.125*IC       $       D=.19+.35*IC
        X=X0+C*DX-D*DY        $       Y=Y0+C*DY+D*DX
        R=AMAX1(ABS(RMIN),ABS(RMIN+DR*RL)) $ R=ALOG10(R)
        IR=INT(ABS(R)) $ IF(R.LT.0.) IR=-(IR+1) $ IR=IR-MOD(IR,3)
        R1=RMIN/10.**IR $ DR1=DR/10.**IR $ R=0.
20      ENCODE(7,101,S)R1 $ CALL SYMBOL(X,Y,.07,S,A,7) $ R1=R1+DR1
        X=X+DX $ Y=Y+DY $ R=R+1. $ IF(R.LE.RL) GO TO 20
        R=(RL-.1*NNC)/2.       $       C=.1+.5*IC
        X=X0+R*DX-C*DY         $       Y=Y0+R*DY+C*DX
        CALL SYMBOL(X,Y,.1,L,ANG,NNC) $ IF(IR.EQ.0) RETURN
        ENCODE(5,102,S) $ CALL SYMBOL(999.,999.,.10,S,ANG,5)
        CALL WHERE(X,Y,A)
        ENCODE(3,103,S) IR  $ CALL SYMBOL(X,Y,.07,S,ANG,3)
101     FORMAT(F7.2)
102     FORMAT(5H  *10)
103     FORMAT(I3)
        RETURN      $      END
```

```
      SUBROUTINE SCALE(DATA,LENGTH,N,K)
      REAL DATA(N),LENGTH,SF(5)
      DATA SF/1.0,2.0,2.5,5.0,10.0/
      DMIN=DMAX=DATA(1)
      DO 10 I=1,N
      IF (DATA(I).LT.DMIN) DMIN=DATA(I)
10    IF (DATA(I).GT.DMAX) DMAX=DATA(I)
      DATA(N+1)=DMIN $ DATA(N+2)=1.0
      IF (LENGTH.LE.0.0.OR.DMAX.EQ.DMIN) RETURN
      RAWSF=(DMAX-DMIN)/LENGTH
      SFEXP=AINT(ALOG10(RAWSF))
      IF (RAWSF.LT.1.0) SFEXP=SFEXP-1.0
      SFMANT=RAWSF*10.0**(-SFEXP)
      DO 20 I=1,5
20    IF (SF(I).GT.SFMANT) GO TO 30
      PRINT*," SCALE: SCALE FACTOR ERROR. " $ RETURN
30    SFNICE=SF(I)*10.0**SFEXP
      ADJMIN=AINT(DMIN/SFNICE)*SFNICE
      IF (ADJMIN.GT.DMIN) ADJMIN=ADJMIN-SFNICE
      IF ((DMAX-ADJMIN)/SFNICE.LT.LENGTH) GO TO 40
      IF (I.LT.5) SFNICE=SF(I+1)*10.0**SFEXP
      IF(I.EQ.5) SFNICE=20.0*10.0**SFEXP
      ADJMIN=AINT(DMIN/SFNICE)*SFNICE
      IF (ADJMIN.GT.DMIN) ADJMIN=ADJMIN-SFNICE
40    DATA(N+1)=ADJMIN $ DATA(N+2)=SFNICE $ RETURN $ END
```

Appendix E

SBC 80/20 Hardware Description (Ref 16)

Memory

There are two types of memory on the SBC 80/20 board: random access memory (RAM) and read-only memory (ROM). Eight INTEL 2113 static RAM devices provide 2048 (2K) X 8-bits of read/write storage. The RAM address space is located from 3800H to 3FFFH by jumper connection 120-121. Four Intel 2708 Erasable and Electrically Programmable Read Only Memory chips provide 4096 (4K) X 8-bits of ROM. The ROM address space is located from 0000H to 0FFFH. A complete memory map is given in Table III, Appendix A. (The functional characteristics of the memories are given in the SBC 80/20 Hardware Reference Manual (Ref 16: Ch 3, 21-29).)

Parallel I/O Interface

Two 8255 Programmable Peripheral Interfaces provide the input ports from the DAS 1128 and the heart rate detector. The remainder of this section describes these interfaces as configured for this specific application. A complete operational summary of the 8255 is available in the hardware reference manual (Ref 16: Ch 3, 51-73).

The 8255 contains three 8-bit ports (A, B, and C). The operating system configures these ports to the strobed input mode (mode 1, control word 'B6H'). This configuration provides for two input ports (A and B) and a control port (C). Each input port contains an input latch to hold the received data while the control port consists of six control bits and two output bits. (The SBC 80/20 modifications listed in the hardware reference manual (Ref 16: Ch 4, 17-18 and 28-29) were accomplished.)

8255 #1 interfaces the DAS 1128 to the SBC 80/20. Port A receives the 8 most significant bits (MSB); Port B receives the 4 least significant bits (LSB) and the 4 multiplexer address bits. The Port C control bits are used as follows:

C_0: INTR_B - interrupt request (not used)
C_1: IBF_B - "high" indicates the data has been loaded into the input latch
C_2: $\overline{\text{STB}}_B$ - "low" loads the data into the input latch
C_3: INTR_A - (not used)
C_4: $\overline{\text{STB}}_A$ - (same as above)
C_5: IBF_A - (same as above)
C_6: (output) - set by control word '0DH'
reset by control word '0CH'
C_7: (output) - set by control word '0FH'
reset by control word '0EH'

The $\overline{\text{EOC}}$ signal from the DAS 1128 provides the $\overline{\text{STB}}_A$ and $\overline{\text{STB}}_B$ signals. Bit C_6 provides the $\overline{\text{STROBE}}$ signal to the DAS 1128; while bit C_7 provides the $\overline{\text{TRIG}}$ signal (see Appendix F). The complete pin assignments for this interface are given in Table IV, Appendix A.

8255 #2 interfaces the heart rate detector to the SBC 80/20. Port A receives the heart rate count; Port B is not used. The Port C control bits are used as follows:

$C_0 - C_2$: (not used)
C_3: INTR_A - (not used)

C_4: $\overline{STB}_A$

C_5: IBF_A

$C_6 - C_7$: (not used)

A strobe signal ($\overline{STB}$) is generated by the heart rate detector when the count is completed (see Appendix G). The complete pin assignments for this interface are given in Table V, Appendix A.

The 8255 I/O port addresses are given in Table VIII; a complete I/O port addressing table is given in the hardware reference manual (Ref 16: Ch 2, 7).

Table VIII

8255 I/O Port Addresses

I/O Device		I/O Port Address (hexadecimal)
8255 #1	Port A	E4
	Port B	E5
	Port C	E6
	Control	E7
8255 #2	Port A	E8
	Port B	E9
	Port C	EA
	Control	EB

Serial I/O Interface

The 8251 USART provides the output port to the Hazeltine 2000 video terminal (the interface connector pin assignments are listed in Table VI, Appendix A). The remainder of this section describes this interface as configured for the IFPDAS II prototype. A complete summary of the 8251 is available in the hardware reference manual (Ref 16: Ch 3, 34-51).

The system software configures the 8251 as an asynchronous receiver/transmitter. A '4EH' mode instruction programs the USART to the asynchronous mode with 1 stop bit, no parity check, 8 transmitted bits, and a baud rate factor of 16X. A '37H' command instruction sets the request-to-send and data-terminal-ready signals high, enables the receive and transmit capabilities, and resets the error flags. Interval timer '2' supplies the baud rate clock (see the following section).

Interval Timers

The 8253 Programmable Interval Timer includes three separate counters (0, 1, and 2). Counter 0 is used as a frequency divider to produce the 225 Hz clock required by the heart rate circuit (see Appendix G). Control word '36H' configures counter 0 as a square wave rate generator. The counter is then loaded with 12ABH (4779) which produces the desired frequency. Counter 1 is used as a real time clock to inform the CPU of every 50 msec interval. Control word '70H' configures counter 1 to interrupt the CPU when the count is complete. The counter is loaded with D1E5H (53,733) which is equivalent to 49.97 msec. The time required for the CPU to handle the interrupt and reset the timer brings the total time between interrupts to 50.00 msec.

Counter 2 is used as a frequency divider to produce the baud rate clock required by the 8251 USART. Control word 'B6H' configures counter 2 as a square wave rate generator. The counter is then loaded with 0038H (56) which produces the desired frequency. A complete summary of the 8253 is contained in the hardware reference manual (Ref 16: Ch 3, 73-87).

Interrupt Controller

The 8259 Programmable Interrupt Controller provides the capability to recognize interrupt requests, and based on that request, to jump to any location in the memory map. This section describes the operation of the 8259 in the IFPDAS II prototype. A complete operational summary of the 8259 is given in the hardware reference manual (Ref 16: Ch 3, 87-110).

The 8259 uses a jump table stored in PROM (03E0H to 03FFH) to pass control to the interrupt handling routine. When the 50 msec timer expires, an interrupt request (IR3) is sent by the timer to the 8259. The 8259 accepts this request and sends an interrupt to the 8080 CPU. After the CPU acknowledges the request, the 8259 "calls" the fourth entry of the jump table which causes a branch to the 50 msec timer interrupt handler (T50MS).

The operating system programs the 8259, during the power-up routine, to accomplish this task. Two initialization command words (ICW) are required to inform the 8259 of the location and length of the jump table. ICW1 (='F6H') and ICW2 (='03H') "tell" the 8259 that the jump table starts at 03E0H and that the call address interval is 4. After the 8259 receives these two words, it is in the normal (fully nested) mode and is ready to operate.

The 8259 is programmed by the CPU to ignore IR2. IR2 is generated by counter 0 which has a special function when the SBC 80/20 monitor is executing (Ref 15:8). Since counter 0 is used for a different purpose in the prototype than in the monitor, the 8080 sends an operational command word (OCW) to the 8259. OCW1 (='04H') masks off IR2 so that this request is never "seen" by the 8259.

The final command sent to the 8259 by the operating system is an end of interrupt (EOI) command word (OCW2 = '20H'). This command resets the in-service bit (IS3) which allows IR3 to request another interrupt.

Appendix F

DAS 1128 Hardware Description (Ref 17)

The DAS 1128 is a complete self-contained miniature high speed data acquisition system which is described in the attached 8-page pamphlet. The module is hard-wired to its IFPDAS II prototype configuration; these modifications are listed in Table IX along with their function. These hard-wired modifications are also reflected in Figure 17, Appendix A. The complete operating characteristics are given in the pamphlet.

Table IX

DAS 1128 Hardwired Modifications

Function	Jumper Connection
16 single-ended inputs	11B to 11T 12B to 2B 17B to 19T 18T to 18B
Full range scale 0 - 5.12 volts	12T to 13B 14T and 14B to 13T 15B to 16B
Full 12 bit operation	28T to DIG GND
Output code: Unipolar Binary	17T to -15 volts 29T (B1) is MSB
Sequentially triggered multiplexer addressing	24B to +5 volts $\overline{\text{STROBE}}$ to 8255 #1 $\overline{\text{TRIG}}$ to 8255 #1
Sequence 0 to 6, then repeat	4 OUT and 2 OUT to external NAND gate Output of NAND gate to 25B
Highest accuracy	CLK TRIM to DIG GND (provides 2.08 microsec/bit conversion time) DLY TRIM to DIG GND ±15 V return to ANA RTN +5 V return to DIG RTN

ANALOG DEVICES

Low Cost, High Speed Data Acquisition Module

DAS1128

FEATURES

Complete Data Acquisition System
12 Bit Digital Output
16 Single or 8 Differential Analog Inputs
High Throughput Rate
Selectable Analog Input Ranges
Versatile Input/Output/Control Format
Low 3 Watt Power Dissipation
Small 3" x 4.6" x 0.375" Module

GENERAL DESCRIPTION

The DAS1128 is a complete self-contained miniature high speed data acquisition system. The compact 3" x 4.6" x 0.375" module provides the designer with an easily implemented solution to the data acquisition problem. It contains an analog input signal multiplexer, a sample-and-hold amplifier, a 12 bit A/D converter, and all of the programming, timing and control circuitry needed to perform the complete data acquisition function.

The DAS1128 is a high performance device which can digitize an analog signal to an accuracy of ±½LSB out of 12 bits, relative to full scale. It has ±8ppm/°C gain temperature coefficient, and the maximum throughput rate can be varied from 50,000 conversions/second for a 12 bit conversion from different analog input channels, to 200,000 conversions/second for a successive 4 bit conversion made on a single channel.

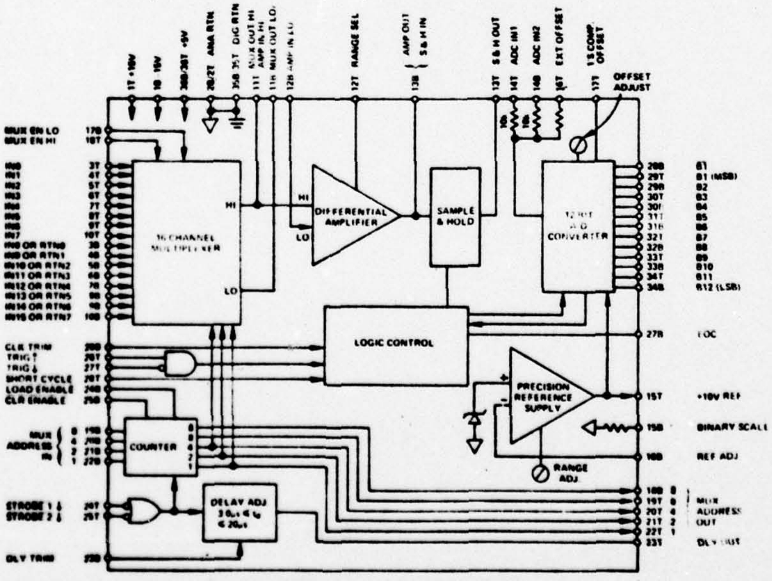


Figure 1. Functional Block Diagram

Route 1 Industrial Park; P.O. Box 280; Norwood, Mass. 02062
Tel: 617/329-4700 TWX: 710/394-6577
West Coast 213/595-1783
Mid-West 312/894-3300
Texas 214/231-5094

SPECIFICATIONS (typical @ +25°C and ±15V unless otherwise noted)

ANALOG INPUTS	
Number of Inputs to Multiplexer	16 Single Ended, 8 True-Differential, 16 Pseudo-Differential
Input Voltage (Full Scale Range)	-10V to +10V, 0V to +10V, -5V to +5V, 0V to +5V, -10.24V to +10.24V, 0V to +10.24V, -5.12V to +5.12V, or 0V to +5.12V.
Maximum Input Voltage	±15V
Input Current (per channel)	5nA max
Input Impedance	$>10^{10}$ ohms
Input Capacitance	10pF for "OFF" channel 100pF for "ON" channel
Input Fault Current (power off or MUX failure)	Internally limited to 20mA
Direct ADC Input Impedance	10kΩ for each input line
ACCURACY[1]	
Resolution	12 Bits
Error Relative to F.S.	±½LSB
Quantization Error	±½LSB
Differential Nonlinearity Error	
● 33kHz throughput rate	±½LSB, 1LSB max
● 50kHz throughput rate	±1LSB
Noise Error	±¼LSB
-FS to +FS Error Between Successive Channel Transitions	±1LSB
TEMP. COEFFICIENTS	
Gain	8ppm/°C, 20ppm/°C max
Offset	5ppm/°C, 15ppm/°C max
Differential Nonlinearity	2.5ppm/°C, 6ppm/°C max
SIGNAL DYNAMICS	
Throughput Rate (12 Bits)	50kHz (max) (includes 5μsecs for MUX and SHA settling time plus 15μsecs for ADC)
MUX Crosstalk ("OFF" channels to "ON" channel)	>80dB down @ 1kHz
Differential Amplifier CMRR	70dB to 1kHz
SHA Acquisition Time to 0.01%	4.5μsec max
SHA Aperture Uncertainty	10nsec
SHA Feedthrough	70dB down @ 1kHz
GITAL INPUT SIGNALS	
Compatibility	Standard DTL/TTL logic levels, 1 unit load/line
MUX Address Inputs (8, 4, 2, 1; Pins 19B through 22B)	Positive true natural binary coding selects channel for random addressing mode. Must be stable for 100nsec after $\overline{\text{STROBE}}$.
MUX ENABLE HI (Pin 18T)	High (logic "1") input enables MUX "HI" output (for inputs 0 through 7)
MUX ENABLE LO (Pin 17B)	High (logic "1") input enables MUX "LO" output (for inputs 8 through 15)
$\overline{\text{STROBE}}$ (Pin 24T or 25T)	Negative going transition (logic "1" to logic "0") updates MUX address register. $\overline{\text{STROBE}}$ 1 must be a logic "1" to enable $\overline{\text{STROBE}}$ 2. $\overline{\text{STROBE}}$ 2 must be at logic "1" to enable $\overline{\text{STROBE}}$ 1.
$\overline{\text{LOAD ENABLE}}$ (Pin 24B)	High (logic "1") input allows next $\overline{\text{STROBE}}$ command to sequentially advance MUX address register. Low (logic "0") input allows next $\overline{\text{STROBE}}$ command to update MUX address register according to external address inputs.
$\overline{\text{CLEAR ENABLE}}$ (Pin 25B)	Low (logic "0") input allows next $\overline{\text{STROBE}}$ command to reset MUX address to channel "0" overriding $\overline{\text{LOAD ENABLE}}$.
TRIGGER (Pin 26T)	Positive going transition (logic "0" to logic "1") initiates A/D conversion (even during conversion); $\overline{\text{TRIGGER}}$ (Pin 27T) must be at logic "0" to allow TRIGGER function.
$\overline{\text{TRIGGER}}$ (Pin 27T)	Negative going transition (logic "1" to logic "0") initiates A/D conversion; Pin 26T (TRIGGER) must be at logic "1" to allow $\overline{\text{TRIGGER}}$ function.
DIGITAL OUTPUT SIGNALS	
Compatibility	Standard DTL/TTL logic levels; 5 unit loads/line.
Parallel Outputs	$\overline{\text{B1}}$, B1 *through* B12
Coding	Natural binary, two's complement, offset binary, or one's complement. Pin selectable.
MUX Addres Outputs ($\overline{8}$, 8, 4, 2, 1; pins 18B, 19T through 22T)	Positive true natural binary coding indicates channel selected.
DELAY OUT (Pin 23T)	Negative going transition (logic "1" to logic "0") occurring normally 5μsecs (adjustable from 3.0μsecs to 20μsecs) after $\overline{\text{STROBE}}$ command initiates A/D conversion automatically when connected to the $\overline{\text{TRIGGER}}$.
$\overline{\text{EOC}}$ (Pin 27B)	High (logic "1") output during A/D conversion.
ADJUSTMENTS & TRIMS	
Offset Adjust	
Internal Adjustment (Externally Accessible)	±10LSB's (min)
Remote External Adjustment (Pin 16T)	±10LSB's (min)
Range Adjust	
Internal Adjustment (Externally Accessible)	±10LSB's (min)
Remote External Adjustment (Pin 16B)	±10LSB's (min)
Clock Trim (Pin 26B)	
Factory Setting (Pin 26B "OPEN")	1.25μs/Bit
External Adjustment Range	1.25μs/Bit to 2.08μs/Bit
Delay Trim (Pin 23B)	
Factory Setting (Pin 23B "OPEN")	3.0μs
External Adjustment Range	3.0μs to 20μs
CONTROLS	
SHORT CYCLE (Pin 28T)	Connect to ground for full 12 bit resolution. Connect to B_n output for resolution to B_{n-1} bits.
Channel Selection Mode (MUX Address Loading Mode)	Random, sequential continuous, and sequential triggered. Pin selectable.
A-D Conversion/Channel-Select Sequences	Normal (input channel remains selected during its A/D conversion) and overlap (next channel selected during A/D conversion). Pin selectable.
Range Select (Pin 12T)	Differential Amplifier gain control: connect to ANA RTN (Pin 2T) for X1 gain; connect to AMP OUT (Pin 13B) for X2 gain. This control is used in FSR selection procedure.
BINARY SCALE (Pin 15B)	Connect to REF ADJ (Pin 16B) to set reference to 10.24V. This control is used in FSR selection procedure, see Table II.
OUTPUT CODING (Pin 17T)	Ground for 1's complement output code; connect to -15VDC for other available codes.
POWER REQUIREMENTS	
+15V ±3%	40mA, 50mA max
-15V ±3%	70mA, 100mA max
+5V ±5%	250mA, 500mA max
Power Supply Sensitivity[2]:	
Gain	±2.0mV/V
Offset	±4.0mV/V
Ref	±0.5mV/V
ENVIRONMENT & PHYSICAL	
Operating Temperature	0° to +70°C
Storage Temperature	-25°C to +85°C
Relative Humidity	Up to 95% non-condensing
Electrical Shielding	RFI & EMI 6 sides (except connector area)
Packaging	Insulated steel cased module 3.00" x 4.60" x 0.375"
PRICE	$295.00 (1 - 9), price includes mating right-angle connector.

---

[1] Warmup time to rated accuracy is 5 minutes.

[2] Specification applies only when tracking +15V and -15V supplies are used, and for slowly occuring variations in power supply voltages.

Specifications subject to change without notice.

THEORY OF OPERATION

A block diagram of the DAS1128 is shown in Figure 1. Analog input signals are applied to the various inputs of the 16 channel CMOS multiplexer. This multiplexer in conjunction with the differential amplifier that follows it, can be configured by the user to accept 16 single ended analog inputs, or 8 fully differential analog inputs. It can also be connected as a 16 channel "pseudo-differential" input device, which permits some of the benefits of differential operation while maintaining a 16 channel input capability.

The differential buffer amplifier is gain programmable by the user via jumpers at the module pins. This feature, along with the selectable reference voltages, permits the user to set up the DAS1128 to operate on any of 8 input voltage ranges. The differential amplifier drives a sample-and-hold amplifier, whose function it is to hold the selected analog input signal at a constant level while the A/D converter is making a conversion.

The A/D converter is a high speed 12 bit successive approximation device that has been designed using the Analog Devices' AD562, 12 bit integrated circuit D/A. The reference voltage for the conversion is supplied by an adjustable precision reference circuit that has a temperature coefficient of 5ppm/°C.

In addition to these basic functional blocks, the DAS1128 also contains all of the clock circuitry necessary to perform the complete data acquisition function. The internal clock can be externally adjusted to provide various throughput rates at different accuracies. Input channel addressing logic is provided, as is the capability to short cycle the A/D converter (i.e. perform conversions of less than 12 bits resolution). It is also possible for the user to adjust the time interval between input channel selection and the commencement of a conversion. The user can thus trade off speed vs. accuracy in the settling time of the multiplexer and sample-and-hold amplifier, as well as speed versus accuracy of the A/D converter.

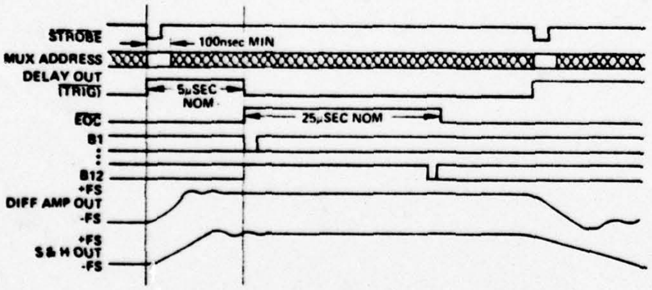


Figure 2. Simplified Timing Diagram, Showing Time-Interval Assignments and Constants.

INPUT CONNECTIONS

As shown in Figure 3, three input configurations can be used. 16 single-ended inputs (3a) can be connected to the multiplexer, all referenced to analog gnd. In the second configuration (3b), the inputs are connected individually as 8 true differential pairs. In this case the differential amplifier is connected "Differentially" with the output of the MUX. Finally, a "Quasi-Differential" connection (3c) can be realized under favorable ground path conditions. In this configuration the differential amplifier Lo terminal is used as the ground return for all sensors. In each of these input schemes, it should be noted that the input multiplexer has been designed to protect itself and signal sources from both overvoltage failure and from fault currents due to power-off loading or MUX failure.

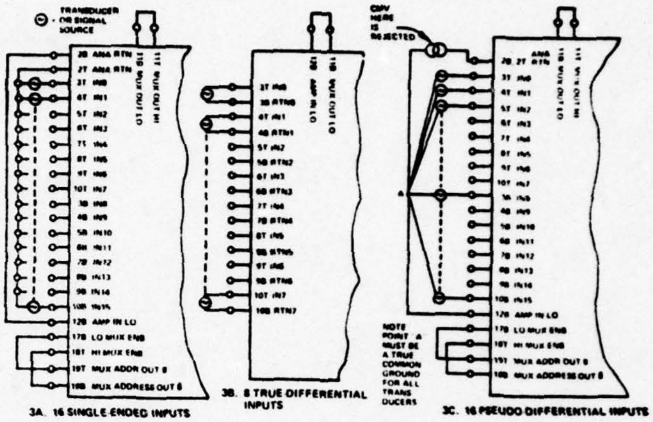


Figure 3. Signal Input Connections for Three Different Configurations.

Full scale range of the DAS1128 may be set by appropriate jumper connections for 8 different ranges: 0 to +10V; 0 to +5V; 0 to +10.24V; 0 to +5.12V; -10 to +10V; -5 to +5V; -10.24 to +10.24V; -5.12 to +5.12V.

Note that 10.24 and 5.12 ranges are commonly used since conversion increments become 5mV/bit, 2.5mV/bit, and 1.25mV/bit.

MUX AND S/H DYNAMICS – OVERLAP MODE

The overlap mode is defined as the ability of MUX to accept a new channel address thereby selecting the next channel to be sampled while the previously acquired sample is being held by the S/H for conversion. The dynamic characteristics of the S/H circuit are shown in Figure 4. Maximum throughput rates are obtainable when a single channel is held at a single address and the channel is sampled repeatedly. In a dynamic condition, data-throughput rates obtainable are shown in Figure 5.

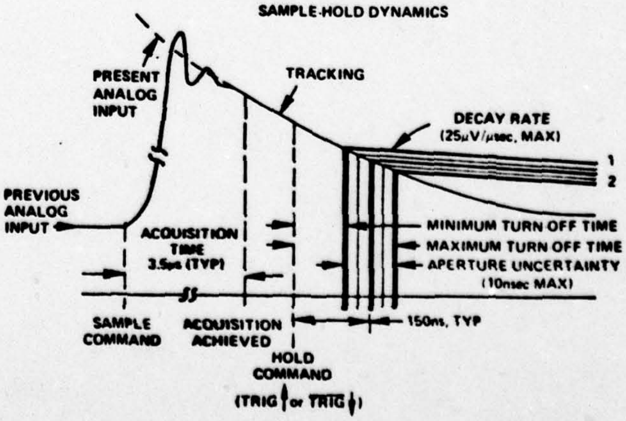


Figure 4. Sample-Hold Parameters Defined and Specified

SHORT CYCLE

It is possible to short cycle the DAS1128, i.e. stop the conversion after less than 12 bits. This can be done by connecting an external jumper between short cycle terminal and one of the output terminals. With shorter cycles the attainable throughput rate increases, see Figure 5. In short cycle operation the TC will decrease proportionately to the number of bits selected. Note the short cycle terminal *must* be grounded for full 12-bit operation.

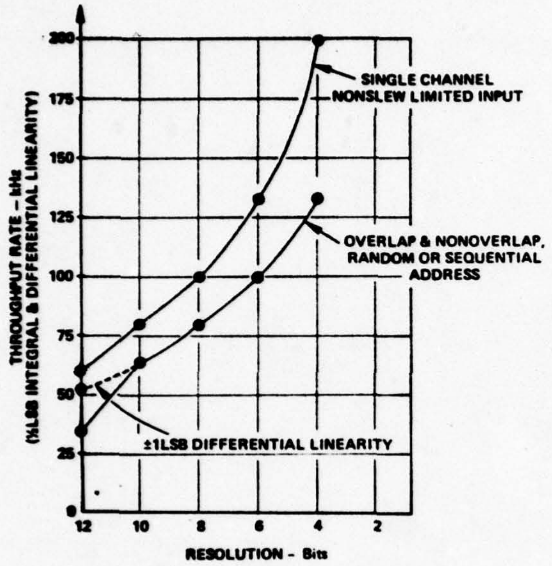
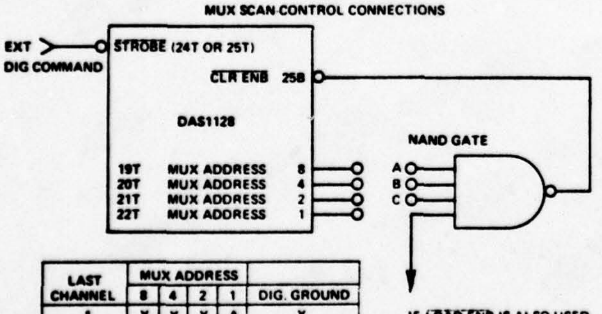


Figure 5. DAS1128 Throughput Rates

MUX ADDRESSING

External terminals have been provided for the address counter. Thus the address counter can be configured to produce the following modes: Continuous sequential scanning (free running), sequential scanning with external step command, abbreviated scan continuously, random channel selection. See Figure 6 and set up procedure for details.



LAST CHANNEL	MUX ADDRESS				
	8	4	2	1	DIG. GROUND
1	X	X	X	A	X
2	X	X	B	X	X
3	X	X	B	A	X
4	X	C	X	X	X
5	X	C	X	A	X
6	X	C	B	X	X
7	X	C	B	A	X
8	C	X	X	X	X
9	C	X	X	A	X
10	C	X	B	X	X
11	C	X	B	A	X
12	B	C	X	X	X
13	B	C	X	A	X
14	A	C	B	X	X
15	X	X	X	X	A

X = NO CONNECTION

IF LOAD ENB IS ALSO USED, CONNECT THIS INPUT TO IT (TO 24B). THIS GIVES LOADING PREFERENCE OVER CLEARING.

NOTE: FOR INITIAL CHANNEL = M, CONNECT GATE OUTPUT TO LOAD ENB (24B) AND CODE CHANNEL IN 8, 4, 2, 1 TO ADDRESS M. HOWEVER, THIS PRECLUDES RANDOM ADDRESSING.

Figure 6. To shorten scanning sequency of multiplexer channels, make the appropriate connections, (as shown in the chart) between an external NAND gate and MUX ADDRESS terminals 19T to 21T.

GROUNDING CONSIDERATIONS

Attention should be given to the methods of connection for electrical returns and voltage reference points. Analog return (ANA RTN) and digital return (DIG RTN) are provided. The following rules should be applied when integrating the DAS1128 into the system.

1. If the ±15V power supply is floating (for optimum analog accuracy), connect its return to ANA RTN (Pin 2B or 2T). If the ±15V power supply is *not* floating, connect its return to DIG RTN (Pin 35T or 36T).
2. Connect the +5V supply return to DIG RTN (Pin 35T or 36T). If this supply also powers additional equipment, run separate, parallel returns to the equipment ground and to DIG RTN (Pin 35T or 36T).
3. To minimize signal grounding problems, single-ended input signals should only be returned to ANA RTN (Pin 2B or 2T). If this is not possible, then connect the input signals in either the "true differential" or "pseudo-differential" configurations (see Figure 3).
4. Connect computer ground to DIG RTN (Pin 35T or 35B). Use heavy wire or ground planes.
5. The computer chassis should be connected to the computer and power supply grounds at only one point.
6. Connect the third-wire ground from main AC power input to the computer power supply return.

GAIN AND OFFSET ADJUSTMENTS

The DAS1128 is calibrated with external gain and offset adjustment potentiometers connected as shown in Figure 7 and 8. The offset adjustment potentiometer has an adjustment range of at least ±10LSB's, and the gain range adjustment potentiometer has an adjustment range of at least ±10LSB's.

Offset calibration is not affected by changes in gain calibration, and should therefore be performed prior to gain calibration. Proper gain and offset calibration requires great care and the use of extremely sensitive and accurate reference instruments. The voltage standard used as a signal source must be very stable. It should be capable of being set to within ±1/10LSB of the desired value at any point within its range.

These adjustments are not made with zero and full scale input signals, and it may be helpful to understand why. An A/D converter will produce a given digital word output for a small range of input signals, the nominal width of the range being one LSB. If the input test signal is set to a value which should cause the converter to be on the verge of switching between two adjacent digital outputs, the unit can be calibrated so that it does switch at just that point. With a high speed convert command rate and a visual display, these adjustments can be performed in a very accurate and sensitive way. *Analog Devices' Conversion Handbook* gives more detailed information on testing and calibrating A/D converters.

OFFSET CALIBRATION

For unipolar operation set the input voltage precisely to +0.0012V and adjust the offset potentiometer until the converter is just on the verge of switching from 000000000000 to 000000000001.

For ±5V bipolar operation set the input voltage precisely to -4.9988V; for ±10V units set it to -9.9976V. Adjust the offset potentiometer, Figure 7, until Offset Binary coded units are just on the verge of switching from 000000000000 to 000000000001 and Two's Complement coded units are just on the verge of switching 100000000000 to 100000000001.

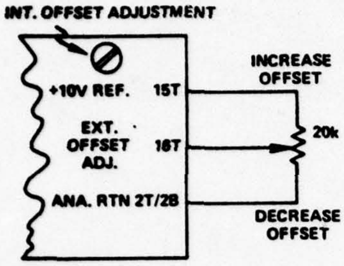
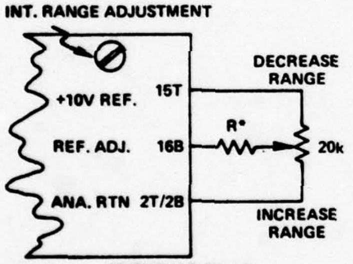


Figure 7. Ext. Offset Adjustment

GAIN CALIBRATION

Set the input voltage precisely to +9.9963V for unipolar operation, +4.9963V for inputs of ±5V or +9.9926V for inputs of ±10V. Note that these values are 1½LSB's less than nominal full scale. Adjust the 20k variable gain resistor, Figure 8, until Binary and Offset Binary coded units are just on the verge of switching from 111111111110 to 111111111111 and Two's Complement coded units are just on the verge of switching from 011111111110 to 011111111111.



*R SHOULD BE AT LEAST 47kΩ AND NO MORE THAN 470kΩ. REDUCING R INCREASES ADJUSTMENT RANGE.

Figure 8. Ext. Ref. Adjustment

CLOCK RATE ADJUSTMENT

The clock rate may be adjusted for best conversion time/accuracy trade-off. The conversion time is varied by means of the external circuitry shown in Figure 9. An open CLK TRIM terminal (Pin 26B) results in 1.25μsec/bit nominal conversion time. A grounded CLK TRIM terminal (for highest accuracy) results in 2.08μsec/bit conversion.

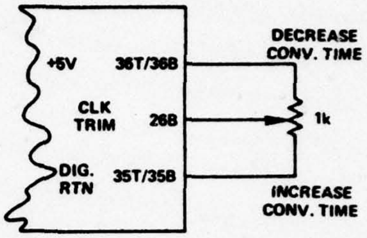


Figure 9. Clock Trim

DELAY TIME ADJUSTMENT

The DLY OUT signal may be adjusted to vary the A/D converter triggering time by means of the external circuitry shown in Figure 10. An open DLY TRIM terminal (Pin 23B) results in a nominal delay time of 3.0μsec. A grounded DLY TRIM terminal (for highest-accuracy) results in 20μsec delay time nominal.

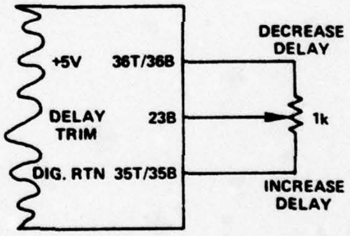


Figure 10. Delay Trim

TABLE I

INPUT CONFIGURATION	ANALOG INPUT CONNECTIONS	ANALOG INPUT RETURN	JUMPER CONNECTIONS
16 Single-Ended Inputs (Figure 3a)	3T thru 10T and 3B thru 10B	All input returns to 2B or 2T	11B to 11T 12B to 2B or 2T 17B to 19T 18T to 18T
8 Differential Inputs (Figure 3b)	3T thru 10T	3B thru 10B	11B to 12B 17B to 18T to "1"
16 Pseudo-Differential Inputs (Figure 3c)	3T thru 10T and 3B thru 10B	Common Input return to 12B	11B to 11T 17B to 19T 18T to 18B

RECOMMENDED SET-UP PROCEDURE

1. Select input configuration, see Table I.
2. Select MUX address mode.
   The method of addressing the multiplexer can be selected by connecting the unit as follows:

   RANDOM. Set Pin 24B ($\overline{\text{LOAD ENB}}$) to logic "0". The next falling edge of $\overline{\text{STROBE}}$ will load the address presented to Pins 19B through 22B (8, 4, 2, 1). The code on these lines must be stable during the falling edge of $\overline{\text{STROBE}}$ plus 100nsec.

   SEQUENTIAL FREE RUNNING. Set to logic "1", Pin 24B ($\overline{\text{LOAD ENB}}$) and 25B ($\overline{\text{CLR ENB}}$). Connect Pin 27B ($\overline{\text{EOC}}$) to Pin 24T ($\overline{\text{STROBE 1}}$). Connect Pin 23T (DLY OUT) to Pin 27T ($\overline{\text{TRIG}}$). Use Pin 26T ($\overline{\text{TRIG}}$) as a run/stop control (i.e. A/D conversion will continue while TRIG is high and will stop while TRIG is low).

   SEQUENTIAL TRIGGERED. Set to logic "1", Pins 24B ($\overline{\text{LOAD ENB}}$) and 25B ($\overline{\text{CLR ENB}}$). Connect Pin 24T ($\overline{\text{STROBE}}$) to external triggering source. The multiplexer address register will automatically advance by one channel whenever a $\overline{\text{STROBE}}$ command is received. The initial channel can be selected by setting Pin 24B ($\overline{\text{LOAD ENB}}$) to logic "0" during only one $\overline{\text{STROBE}}$ command. The multiplexer address will then be determined by the logic levels on Pins 19B through 22B (the external MUX address lines). Channel "0" can be selected as the initial channel by setting Pin 25B ($\overline{\text{CLR ENB}}$) to logic "0" during only one $\overline{\text{STROBE}}$ command. The final channel can be selected by following the procedure presented in Figure 6.
3. Select A-D conversion/channel select sequence (see Figure 5).
   (1) NORMAL (input channel remains selected during its A/D conversion). Connect Pin 23T (DLY OUT) to Pin 27T ($\overline{\text{TRIG}}$).

   (2) OVERLAP (next channel is selected during A/D conversion). Connect Pin 27B ($\overline{\text{EOC}}$) to TTL compatible inverter input. Connect inverter output to Pin 24T ($\overline{\text{STROBE}}$). Connect Pin 23T (DLY OUT) to Pin 27T ($\overline{\text{TRIG}}$). Adjust the delay to at least 4μsec greater than $\overline{\text{EOC}}$, 20μsec max (see Figure 10). The signal on Pin 26T (TRIG) serves as RUN/STOP control.

   (3) REPETITIVE SINGLE CHANNEL. After selecting the input channel to be repetitively sampled (see MUX ADDRESS MODE, above), set Pin 27T ($\overline{\text{TRIG}}$) to logic "0". Connect Pin 26T (TRIG) to a triggering source. Conversion process is initiated by positive edge of TRIG command.
4. Select output resolution.
   a. Full 12 bit resolution: connect Pin 28T (SHT CYC) to Pin 35B (DIG RTN).
   b. Bn ($Bn < 12$) bit resolution: connect Pin 28T to the output pin for Bn + 1.
5. Select optimum throughput rate.
   The system clock frequency and the STROBE to TRIG delay (if used) can be trimmed to optimize the accuracy/throughput rate trade-off. See Figures 9 and 10.
6. Select input voltage full scale range. See Table II.
7. Select output digital coding. See Table III.

TABLE II

FOR FULL SCALE RANGE OF:	MAKE THE FOLLOWING CONNECTIONS
0 to +10V	12T to 2T; 14T to 14B to ADC Source*.
0 to +10.24V	same as 0 to +10V, *plus* 15B to 16B.
0 to +5V	12T to 13B; 14T and 14B to ADC Source*.
0 to +5.12V	same as 0 to +5V, *plus* 15B to 16B
-10V to +10V	12T to 2T; 14T to 15T; and 14B to ADC Source*.
-10.24V to +10.24V	same as -10V to +10V, *plus* 15B to 16B
-5V to +5V	12T to 13B; 14T to 15T and 14B to ADC Source*.
-5.12V to +5.12V	same as -5V to +5V, *plus* 15B to 16B.

*ADC Source is usually Sample and Hold Output (13T), but may be any signal source including Diff. Amp. Output (13B) if Sample and Hold is not desired.

TABLE III

OUTPUT CODE	CONNECTIONS
Unipolar Binary	Connect 17T to -15V Use 29T (B1) for MSB
2's Complement	Connect 17T to -15V Use 28B ($\overline{\text{B1}}$) for MSB
Offset Binary	Connect 17T to -15V Use 29T (B1) for MSB
1's Complement	Connect 17T to 2B Use 28B ($\overline{\text{B1}}$) for MSB

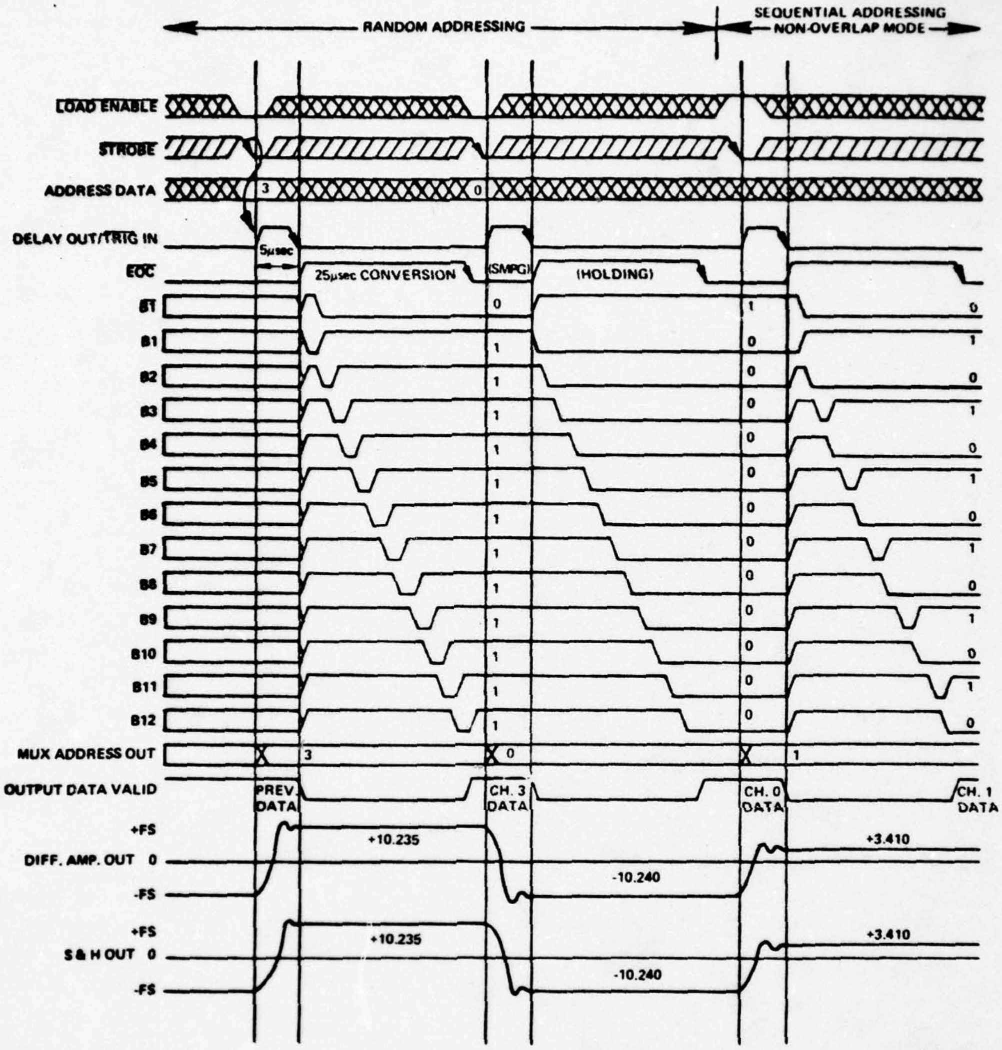


Figure 11. Timing for Non-Overlap Operation in Both Random and Sequential Addressing Modes. For Status Keys and Signal Condition Data, Refer to Box Below.

SIGNAL CONDITIONS
AND STATUS KEYS
FOR FIGURES 11 AND 12.

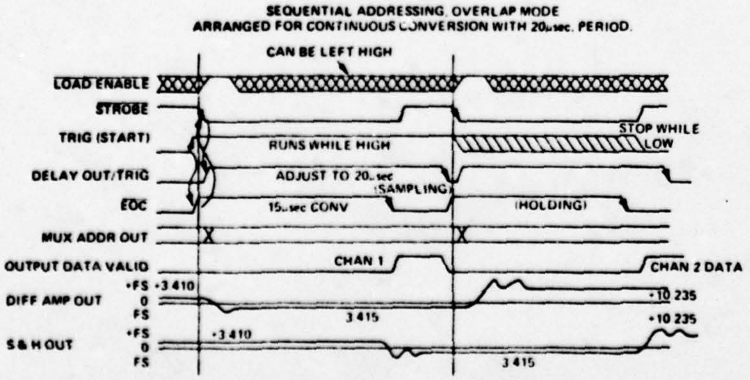


Figure 12. Timing Diagram for Overlap Operation in the Sequential Addressing Mode. For Status Keys and Signal Condition Data, See Box at Right.

CH. 2 = -3.415V CODE 010 101 010 101
CH. 3 = +10.235V CODE 111 111 111 111
CH. 0 = -10.240V CODE 000 000 000 000
CH. 1 = +3.410V CODE 101 010 101 010

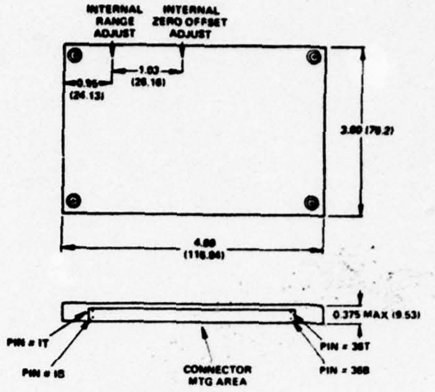
ADC SET UP FOR ±10.24V. INPUT, OFFSET BINARY. (FOR TWO'S COMPLEMENT, USE B̄1 FOR M.S.B.)

KEY	INPUTS	OUTPUTS
XXX	May change	Don't know
///	May change 0 to 1	Changes 0 to 1
\\\	May change 1 to 0	Changes 1 to 0
OR	Must be stable	Will be stable

Outline Drawings and Pin Designations

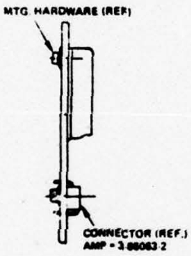
DAS1128 Connector Pin Diagram

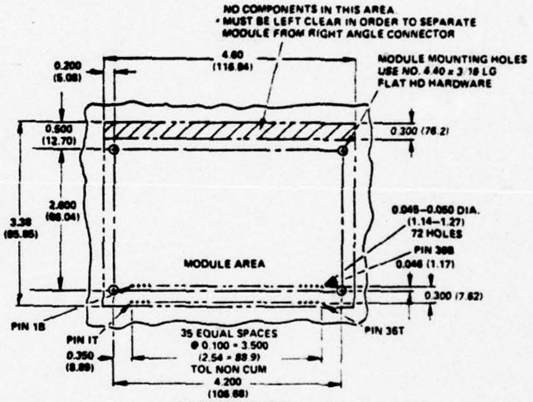
+15V		1T	1B	-15V	
ANA RTN		2T	2B	ANA RTN	
CH 0 IN		3T	3B	CH 8 IN	(CH 0 RTN)
CH 1 IN		4T	4B	CH 9 IN	(CH 1 RTN)
CH 2 IN		5T	5B	CH 10 IN	(CH 2 RTN)
CH 3 IN		6T	6B	CH 11 IN	(CH 3 RTN)
CH 4 IN		7T	7B	CH 12 IN	(CH 4 RTN)
CH 5 IN		8T	8B	CH 13 IN	(CH 5 RTN)
CH 6 IN		9T	9B	CH 14 IN	(CH 6 RTN)
CH 7 IN		10T	10B	CH 15 IN	(CH 7 RTN)
MUX HI OUT		11T	11B	MUX LO OUT	
RANGE SEL		12T	12B	AMP IN LO	
S & H OUT		13T	13B	AMP OUT	
ADC IN 1		14T	14B	ADC IN 2	
+10V REF		15T	15B	BINARY SCALE	
EXT OFFSET		16T	16B	REF ADJ	
OUTPUT CODING		17T	17B	ENABLE LO	
ENABLE HI		18T	18B	$\overline{8}$ OUT	
8 OUT	MUX ADDRESS LINES	19T	19B	8 IN	MUX ADDRESS LINES
4 OUT		20T	20B	4 IN	
2 OUT		21T	21B	2 IN	
1 OUT		22T	22B	1 IN	
DLY OUT		23T	23B	DLY TRIM	
$\overline{\text{STROBE 1}}$		24T	24B	$\overline{\text{LOAD ENB}}$	
$\overline{\text{STROBE 2}}$		25T	25B	CLR ENB	
TRIG		26T	26B	CLK TRIM	
$\overline{\text{TRIG}}$		27T	27B	$\overline{\text{EOC}}$	
SHT CYC		28T	28B	$\overline{\text{B1}}$ OUT	
B1 OUT		29T	29B	B2 OUT	
B3 OUT		30T	30B	B4 OUT	
B5 OUT		31T	31B	B6 OUT	
B7 OUT		32T	32B	B8 OUT	
B9 OUT		33T	33B	B10 OUT	
B11 OUT		34T	34B	B12 LSB OUT	
DIG RTN		35T	35B	DIG RTN	
+5V		36T	36B	+5V	



BASIC MODULE CONFIGURATION

Dimensions shown in inches and (mm).

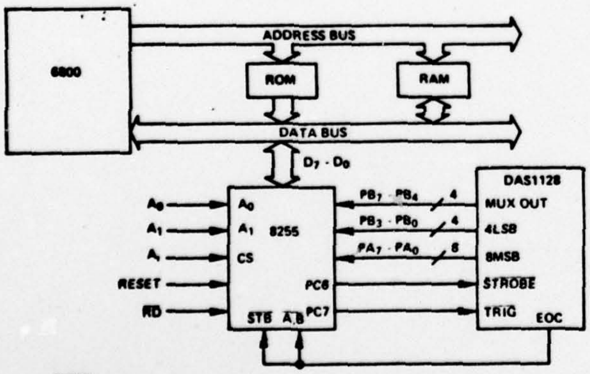




P.C. BOARD LAYOUT

Typical Applications

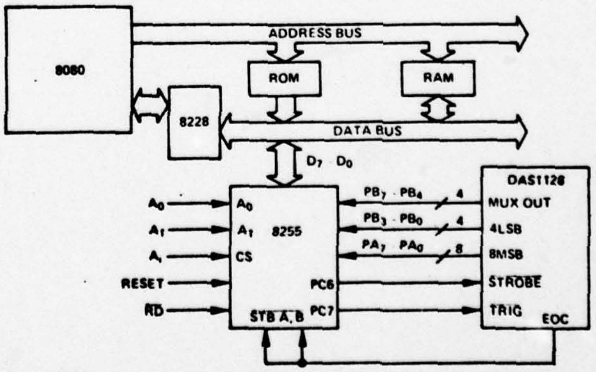
DAS1128 WITH MOTOROLA 6800



NOTE:

1. 8255 USED IN MODE 1 (STROBED I/O)
2. PC6 INDEXES MUX TO DESIRED CHANNEL
3. CS TO A_i (WHERE, A_i IS AN ADDRESS BIT OTHER THAN A_0 OR A_1)
4. PC7 INITIATES CONVERSION
5. EOC STROBES IN DATA AND MUX INFO
6. 8255 SHOWN, HOWEVER 6820 CAN ALSO BE USED

DAS1128 WITH INTEL 8080



NOTE:

1. 8255 USED IN MODE 1 (STROBED I/O)
2. CS TO A_i (WHERE, A_i IS AN ADDRESS BIT OTHER THAN A_0 OR A_1)
3. PC6 INDEXES MUX TO DESIRED CHANNEL
4. PC7 INITIATES CONVERSION
5. EOC STROBES IN DATA AND MUX INFO

C386-8-5/76

PRINTED IN U.S.A.

Appendix G

Sensor Interfaces

This appendix discusses the three circuits that were designed and built to interface the oxygen partial pressure sensors, the accelerometers, and the heart rate detector to the IFPDAS II prototype.

OM11 Sensor Interface (Ref 18)

The circuit in Figure 21 interfaces both Beckman OM11 polarographic sensors to the IFPDAS II prototype. The OM11 sensor is biased to -740 mvolts and produces a current (approximately 2 microAmps for air at room temperature) that is proportional to the oxygen partial pressure of the gas that surrounds the sensor. An amplifier converts this current source to a voltage output. The output signal ranges from 0 to 5.0 volts corresponding to oxygen partial pressures of 0 to 760 mm Hg; the response time for the sensor is 800 msec when exposed to a pure oxygen source.

The D_1-R_1-R_2-R_3 combination acts as a voltage regulator and divider, reducing the -15 volts to the necessary -740 mvolt sensor bias. The sensor current output is then amplified through amplifier A_1 to produce 1.0 volts (152 mm Hg). The C_1-R_5 combination determines the amplifier gain and provides low pass filtering with a 5 Hz cutoff frequency. Temperature compensation for the sensor is provided by a built-in 10K thermistor (R_7) which, in parallel with R_8, automatically adjusts the gain for changing sensor temperature. Calibration of the

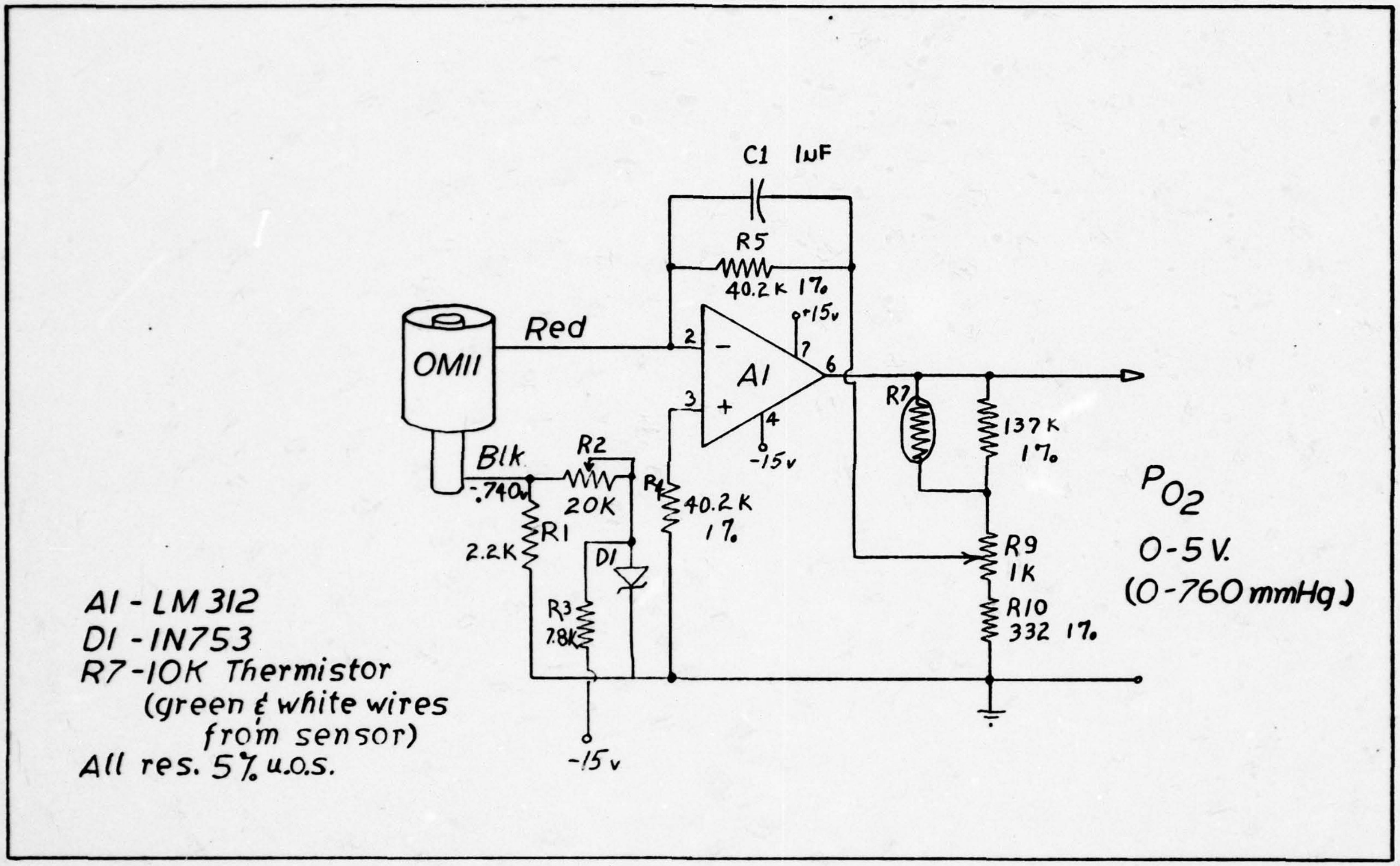


Fig. 21. OM11 Interface Circuit

interface is accomplished using a 100% oxygen source and adjusting potentiometer R_9 for a signal output of 5.0 volts.

If faster sensor response time is necessary, the circuit of Figure 22 can be used to compensate the sensor. This circuit was provided by Beckman Instruments (Ref 19) and it reduces the sensor response time to 100 msec. Power consumption of this circuit can be minimized by utilizing all UA776 amplifiers which can be biased for microwatt power consumption by appropriately selecting the bias resistors (Ref 20: Ch 8, 458-466).

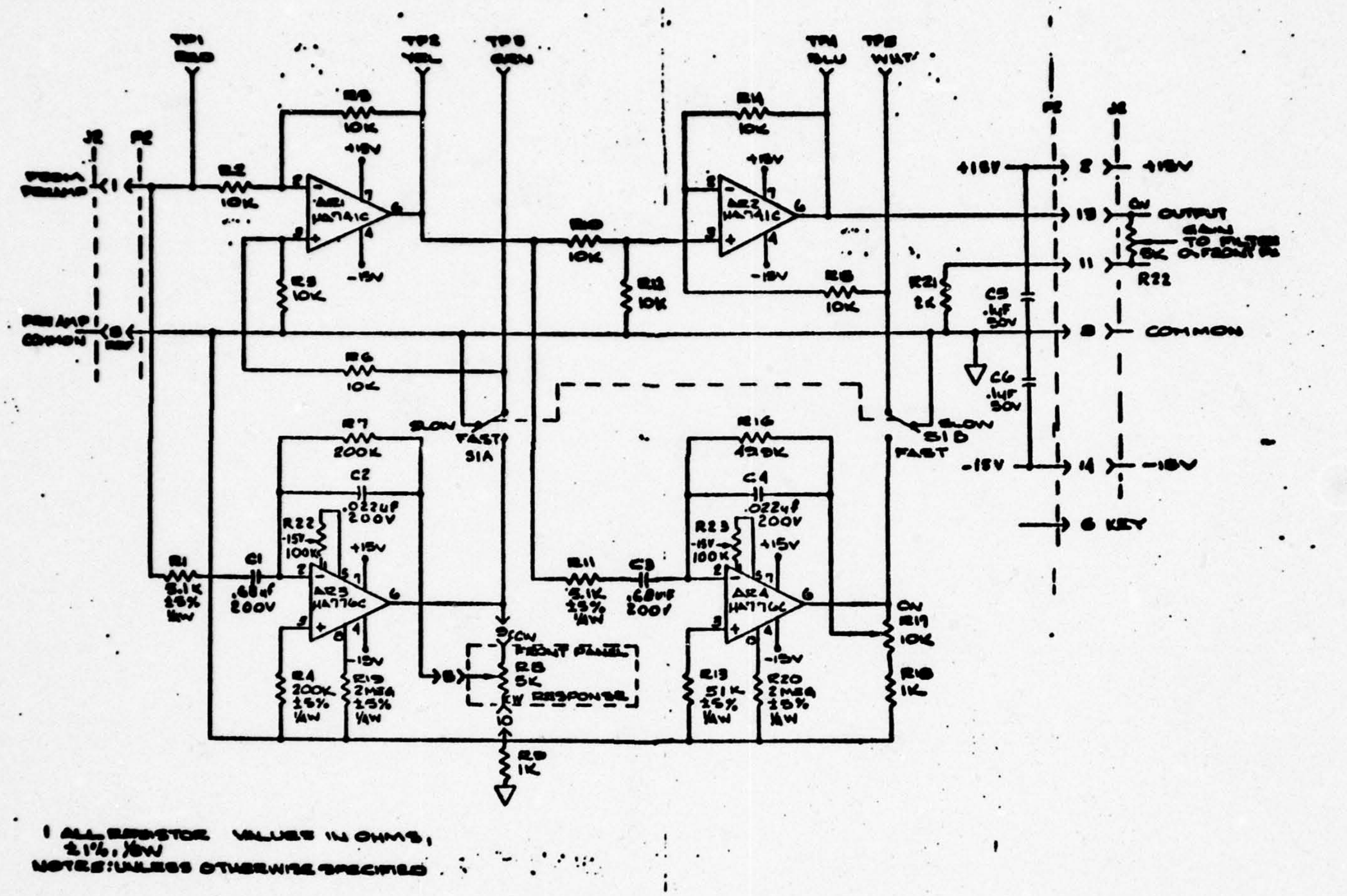


Fig. 22. OM11 Compensator Circuit

Accelerometer Interfaces

The Statham F-15-340 accelerometer interface circuit is shown in Figure 23.

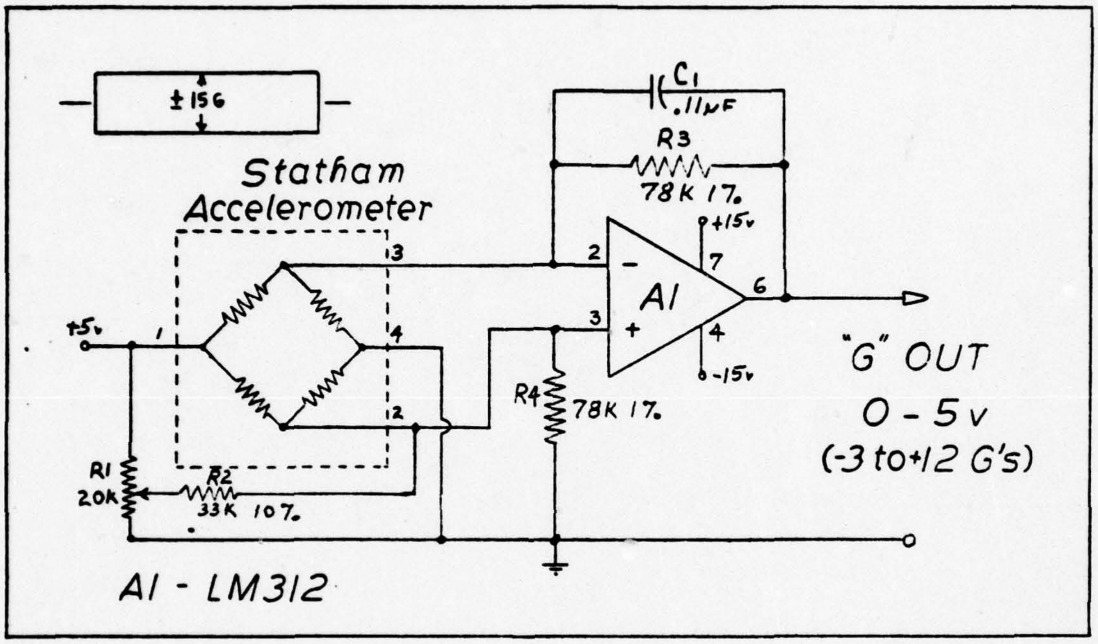


Fig. 23. Accelerometer Interface Circuit

The output range of the circuit is 0 – 5 volts corresponding to -3 to +12 G's. Resistors R_1 and R_2 provide a 1.0 volt (3 G) offset of the bridge. With the accelerometer oriented as shown, R_1 is adjusted to provide a 1.33 volt (1 G) output signal. Feedback network C_1-R_3 determines the amplifier gain; and capacitor C_1 provides filtering to reduce the effects of transient G's. The filter cutoff frequency is approximately 20 Hz.

The output of the amplifier can be tested by changing the orientation of the accelerometer about the longitudinal axis and observing the corresponding signal outputs. With a 90° clockwise rotation, the probe senses a zero-G condition and the amplifier output signal is 1.0 volts. With an additional 90° rotation, the probe senses a negative-one-G condition and the output of the amplifier is .67 volts.

Heart Rate Detector

The heart rate detector consists of an ECG amplifier, R-wave detector, and digital interval counter. The analog portions are shown in Figure 24; the digital portion is shown in Figure 25.

The ECG amplifier increases a 1 mvolt ECG signal to a minimum of 1.5 volts. The differential ECG (leads 1 and 2) is input to amplifiers A_1 and A_2, and lead 3 is used as a reference to reduce the common mode voltage. Feedback networks R_1-C_1 and R_2-C_2 provide the gain and high frequency filtering. Potentiometer R_4 allows the gain of the differential input stage to be varied between 30 and 100.

The amplifier signal is capacitively-coupled to amplifier A_3 through C_3 and C_4. This helps eliminate DC baseline shifts. Amplifier A_3 provides final signal amplification with a differential mode gain of 50. The ECG signal is then filtered through a double-pole, low pass, active filter using amplifier A_4. The filter bandwidth is 60 Hz and its low frequency gain is unity.

Resistors R_1, R_2, and R_7 to R_{10} were chosen within .5% tolerence to reduce the common mode amplifier gain. Resistors R_5, R_6, R_{11}, and R_{15} set bias currents for low power (500 microwatts) operation of the MC1776 operational amplifiers.

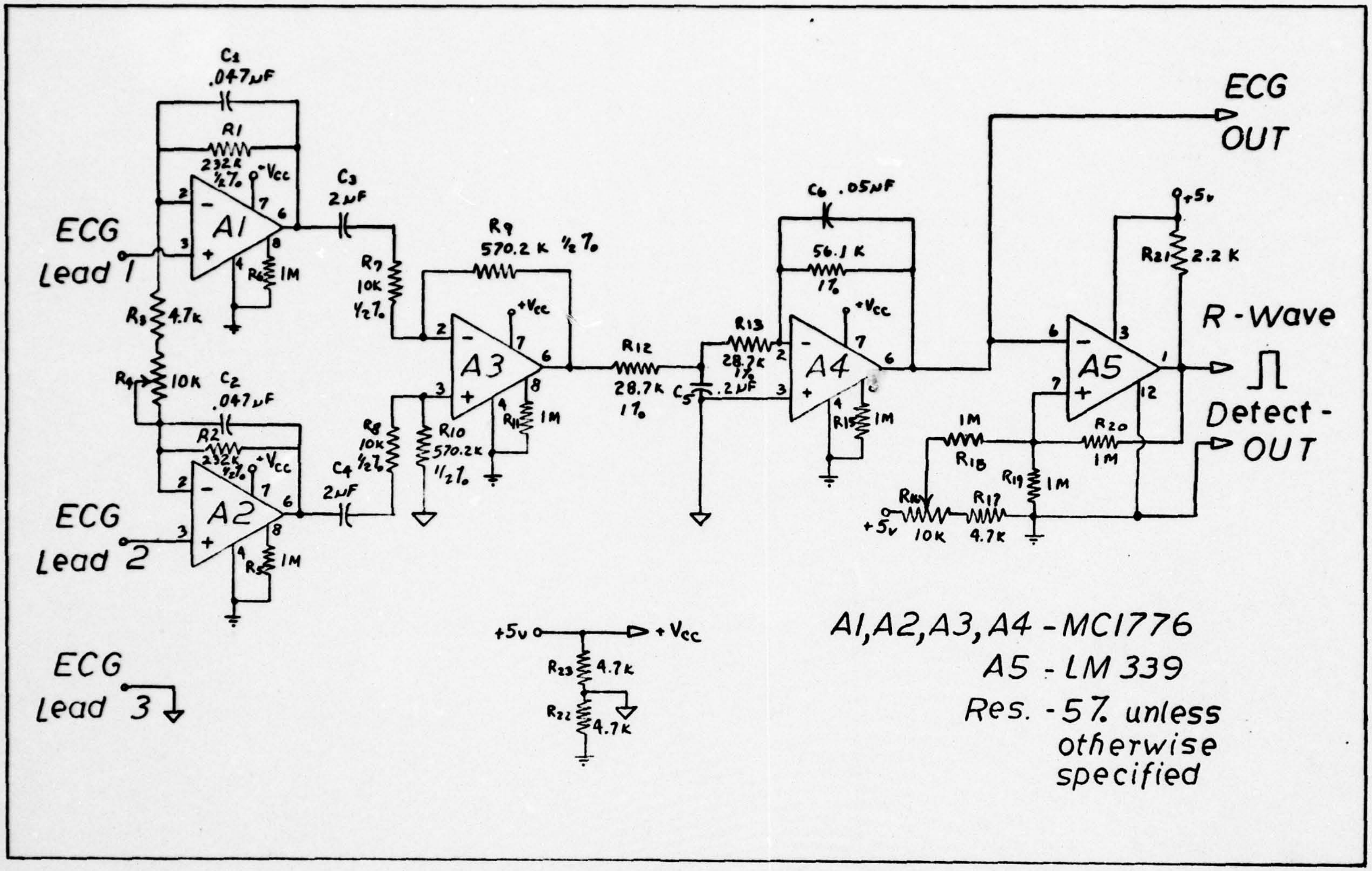


Fig. 24. ECG Amplifier and R-wave Detector

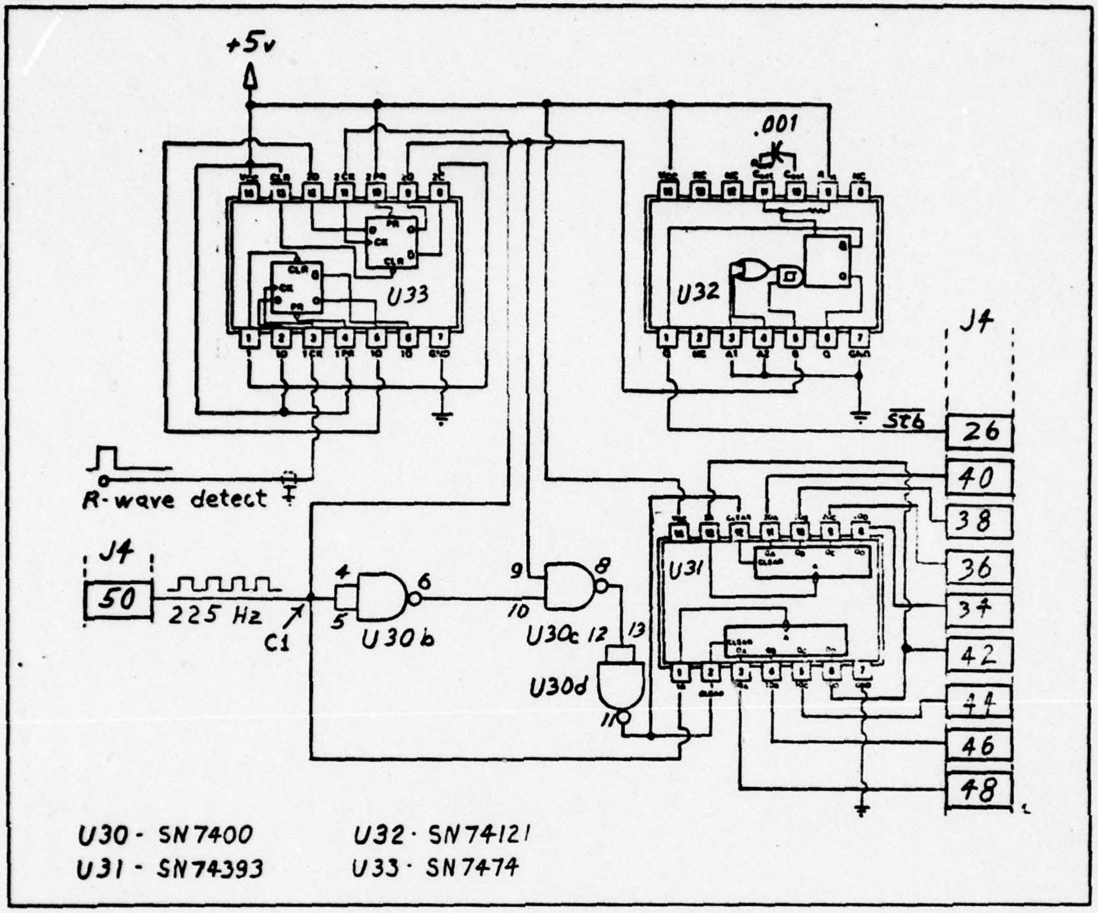


Fig. 25. Digital Interval Counter

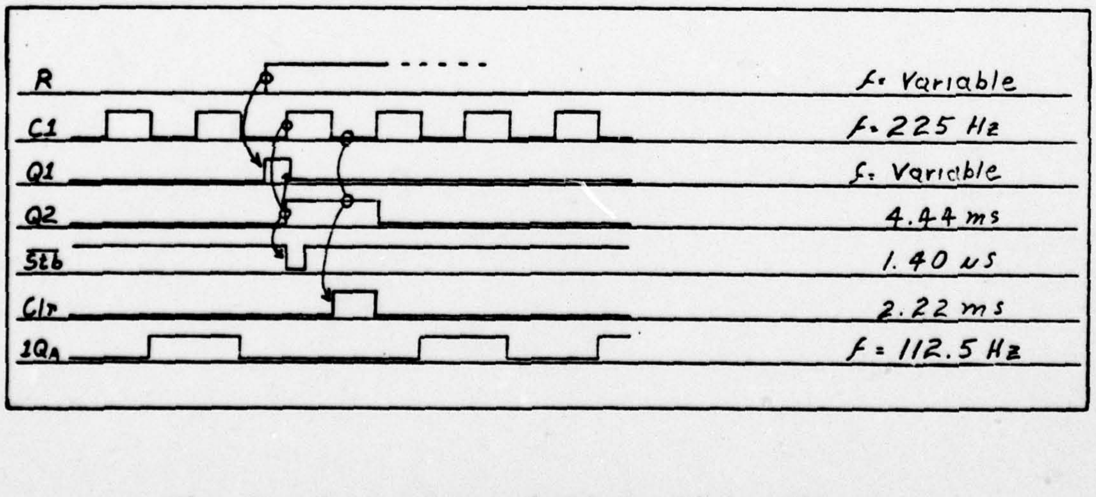


Fig. 26. R-wave Interval Counter Timing Diagram

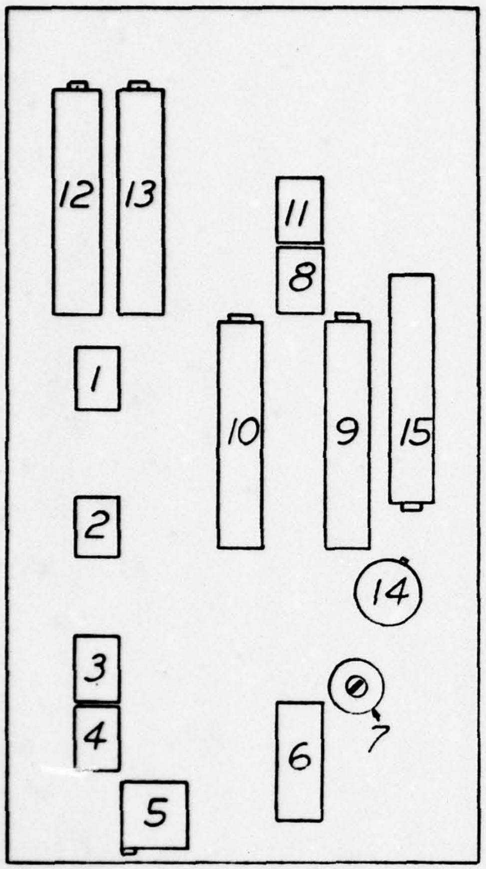
The filtered ECG signal is input to an inverting comparitor (A_5). A reference voltage, adjustable by potentiometer R_{16}, is compared to the ECG input and a pulse is output when the peak of the ECG is above the reference voltage. This pulse is then output to the digital portion of the circuit.

The detector circuit is adjusted by monitoring the ECG on a strip chart or oscilloscope. The amplifier gain is adjusted by R_4 to obtain an ECG peak magnitude of 2 volts. The threshold voltage can be adjusted by R_{16} to detect all of the R-waves without mis-firing on the noise.

The output of the R-wave detector is input to the digital counting circuitry (Figure 25). This circuit counts the number of 4.44 msec (1/225 Hz) periods between detected R-waves. The dual-D flip-flop configuration (U33) shapes the variable width R-wave pulse to a single 4.44 msec pulse. This pulse is input to a monostable multivibrator (U32) to generate a 1.40 microsec $\overline{\text{STB}}$ for the 8255. The 4.44 msec pulse is also gated to clear the counter prior to subsequent counting pulses. (A timing diagram of this sequence is given in Figure 26.)

The oxygen and accelerometer circuits are biased at ± 15 volts so that the DAS 1128 power supply can be used. A +5 volt power supply can be used with low power operational amplifiers (i.e., MC1776 or LM312) by splitting the +5 volt power supply to provide the necessary ± 2.5 volts.

Two OM11 sensor interfaces, the accelerometer interface, and the heart rate detector are contained on a 2.75" x 4.75" component board which is housed in a 3" x 4" x 5" aluminum box. The amplifier and potentiometer orientation is shown in Figure 27. The interface box external connections are shown in Figure 28. The pin assignments for the 25-pin interface connector are given in Table X.



Board#	Part#	Function
Ckt:	Heart rate	
1	A1	
2	A2	
3	A3	
4	A4	
5	R4	ECG Gain adj.
6	A5	
7	R16	Threshold adj.
Ckt:	PO2IN	
8	A1	
9	R9	Gain adj.
10	R2	Bias adj.
Ckt:	PO2OUT	
11	A1	
12	R9	Gain adj.
13	R2	Bias adj.
Ckt:	Acceleration	
14	A1	
15	R1	Offset adj.

Fig. 27. Chassis Board Layout and Part Number/Function

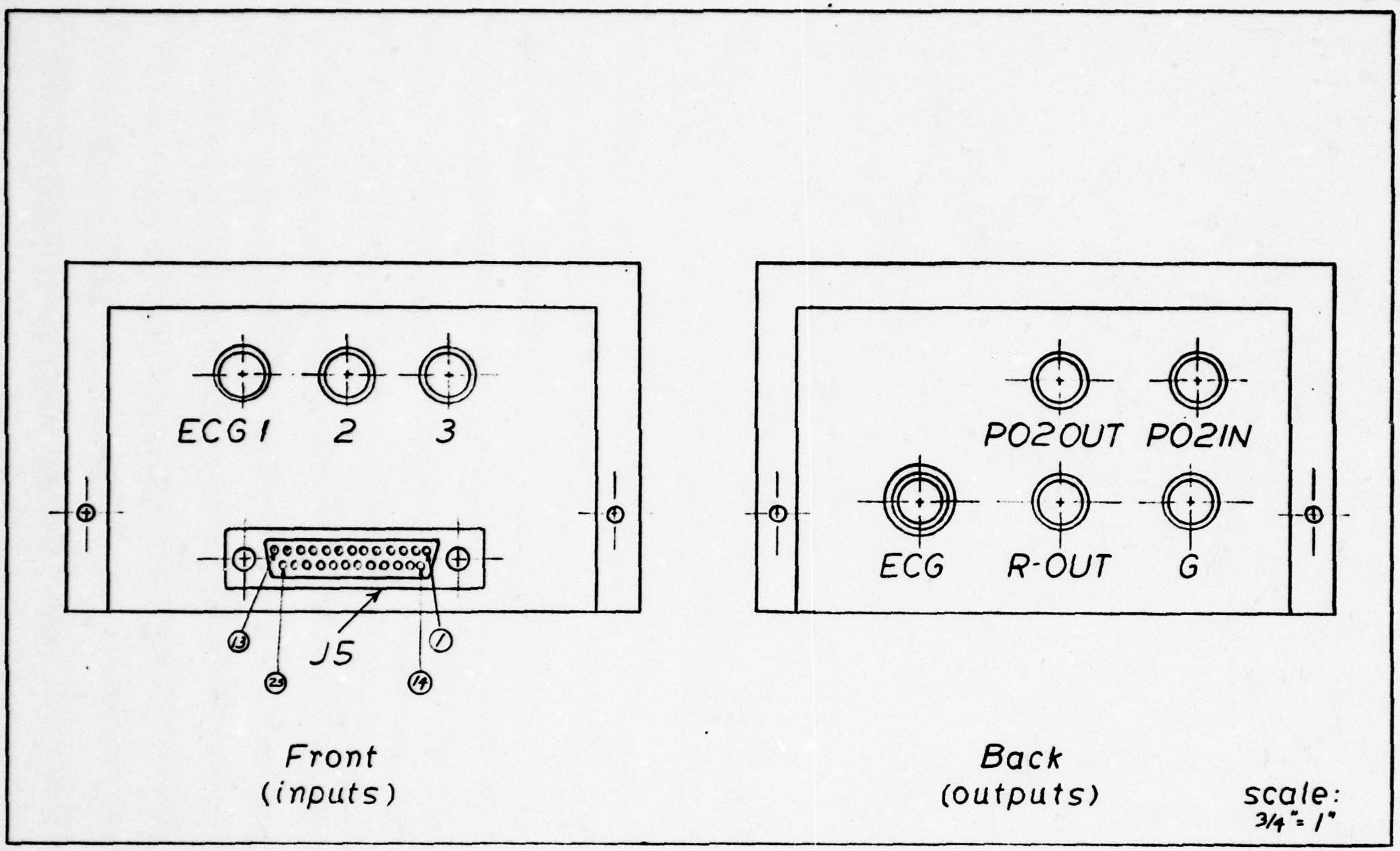


Fig. 28. External Chassis Interface Connections

Table X

Chassis Pin Connections

External Connection	Pin #	Chassis Connection
N/C	1	N/C
thermistor (white)	2	to PO2IN amplifier
OM11 #1 thermistor (green)	3	
(PO2IN) cathode (red)	4	
anode (black)	5	
N/C	6	N/C
N/C	7	N/C
N/C	8	N/C
N/C	9	N/C
accelerometer pin 2	10	to accelerometer amplifier
accelerometer pin 3	11	
+15 VDC (red)	12	+15 volt bus
-15 VDC (white)	13	-15 volt bus
thermistor (white)	14	to PO2OUT amplifier
OM11 #2 thermistor (green)	15	
(PO2OUT) cathode (red)	16	
anode (black)	17	
N/C	18	N/C
N/C	19	N/C
OM11 shields	20	GND
N/C	21	GND
accelerometer pin 4	22	GND
GND (black)	23	GND
accelerometer pin 1	24	+5 volt bus
+5 VDC (blue)	25	+5 volt bus

Appendix H

Magnetic Bubble Memory Interface Diagrams

These diagrams show the interfacing required to test the bubble memory with the prototype. The diagrams were supplied by Gerald Cox of Texas Instruments, Inc., Dallas, Texas.

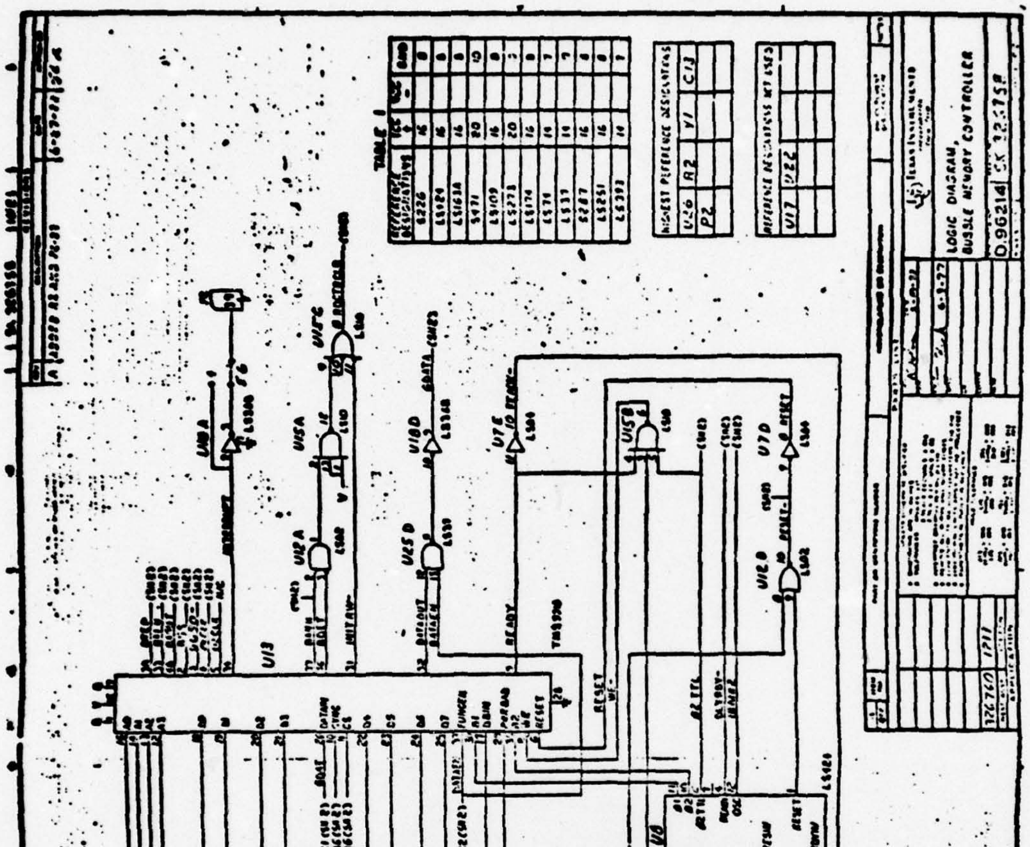


Fig. 29. Logic Diagram, Bubble Memory Controller (Page 1)

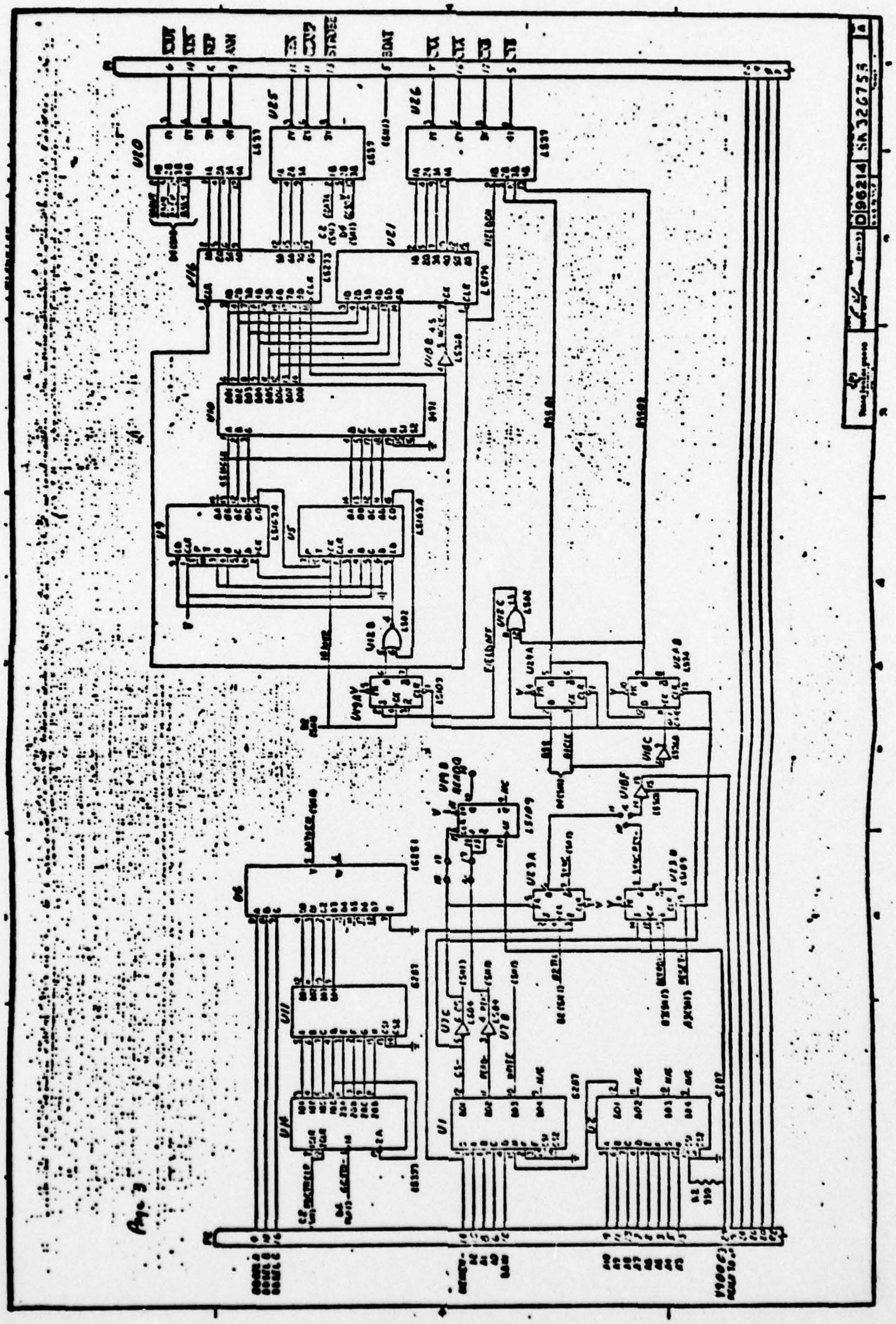


Fig. 30. Logic Diagram, Bubble Memory Controller (Page 2)

THIS PAGE IS BEST QUALITY PRACTICABLE FROM COPY FURNISHED TO DDC

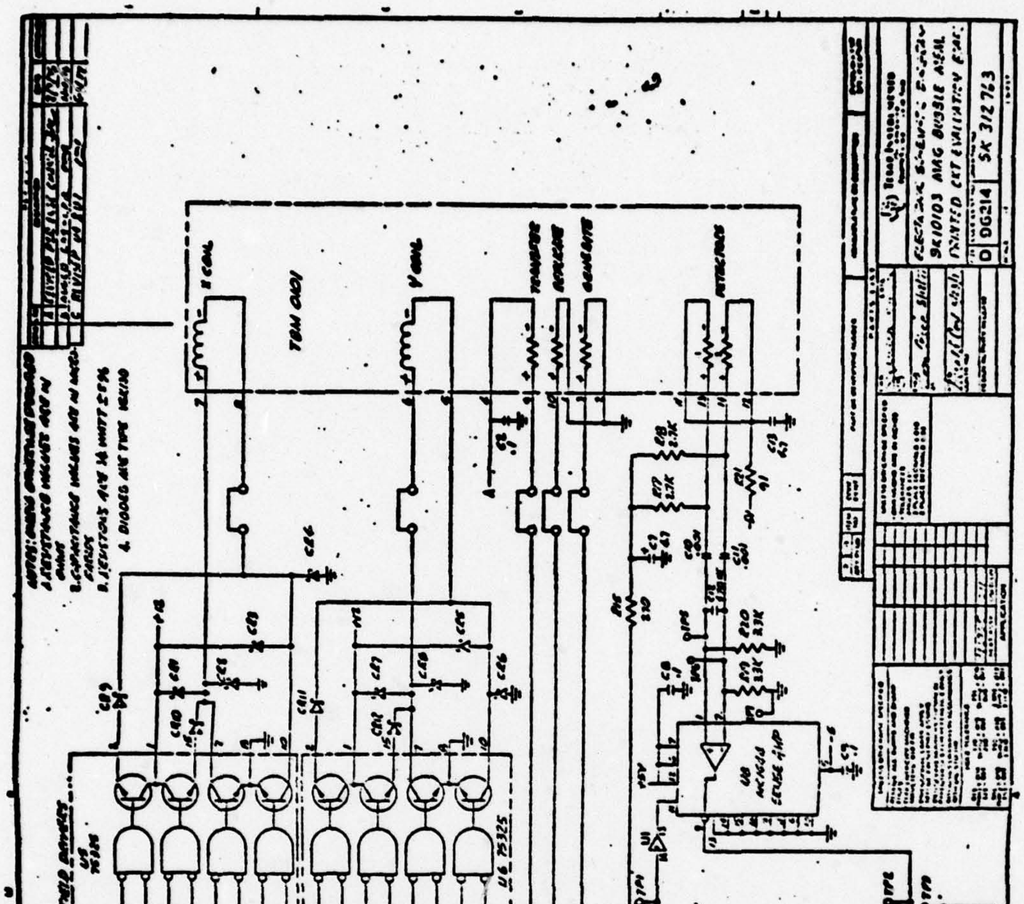
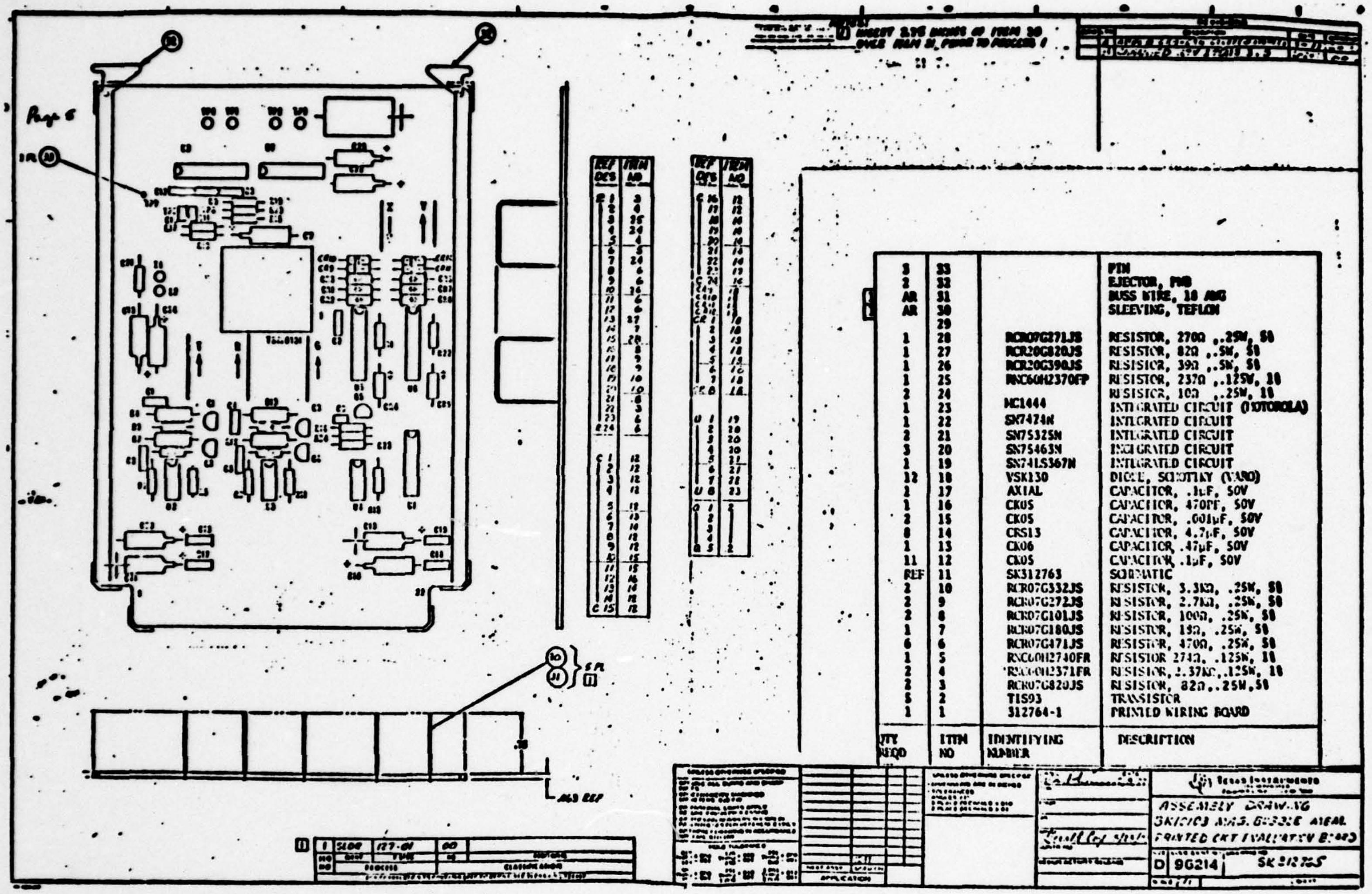


Fig. 31. Magnetic Bubble Memory Printed Circuit Evaluation Board (Page 1)



QTY REQD	ITEM NO	IDENTIFYING NUMBER	DESCRIPTION
3	33		PIN
2	32		EJECTOR, PWB
AR	31		BUSS WIRE, 18 AWG
AR	30		SLEEVING, TEFLON
	29		
1	28	RCR07G271JS	RESISTOR, 270Ω, .25W, 5%
1	27	RCR20G820JS	RESISTOR, 82Ω, .5W, 5%
1	26	RCR20G390JS	RESISTOR, 39Ω, .5W, 5%
1	25	RNC60H2370FP	RESISTOR, 237Ω, .125W, 1%
2	24		RESISTOR, 10Ω, .25W, 1%
1	23	MC1444	INTEGRATED CIRCUIT (MOTOROLA)
1	22	SN7474N	INTEGRATED CIRCUIT
2	21	SN75325N	INTEGRATED CIRCUIT
3	20	SN75463N	INTEGRATED CIRCUIT
1	19	SN74LS367N	INTEGRATED CIRCUIT
12	18	VSK130	DIODE, SCHOTTKY (VARO)
2	17	AXIAL	CAPACITOR, .1µF, 50V
1	16	CK05	CAPACITOR, 470PF, 50V
2	15	CK05	CAPACITOR, .001µF, 50V
8	14	CRS13	CAPACITOR, 4.7µF, 50V
1	13	CK06	CAPACITOR, .47µF, 50V
11	12	CK05	CAPACITOR, .1µF, 50V
REF	11	SK312763	SCHEMATIC
2	10	RCR07G332JS	RESISTOR, 3.3KΩ, .25W, 5%
2	9	RCR07G272JS	RESISTOR, 2.7KΩ, .25W, 5%
2	8	RCR07G101JS	RESISTOR, 100Ω, .25W, 5%
1	7	RCR07G180JS	RESISTOR, 18Ω, .25W, 5%
6	6	RCR07G471JS	RESISTOR, 470Ω, .25W, 5%
1	5	RNC60H2740FR	RESISTOR 274Ω, .125W, 1%
2	4	RNC60H2371FR	RESISTOR, 2.37KΩ, .125W, 1%
2	3	RCR07G820JS	RESISTOR, 82Ω, .25W, 5%
5	2	TIS93	TRANSISTOR
1	1	312764-1	PRINTED WIRING BOARD

Fig. 32. Magnetic Bubble Memory Printed Circuit Evaluation Board (Page 2)

Appendix I

Oxygen Consumption Calculations for Single Breath Analysis

Oxygen consumption is measured by analyzing the mass of the gases of inspiration and expiration. When laboratory conditions permit, precise measurement of inspired and expired flow volumes can be made, and their gaseous contents can be accurately analyzed. Oxygen consumption is then calculated by subtracting the quantity of oxygen exhaled from the quantity of oxygen inhaled (Ref 21:681-685).

Since the IFPDAS II is used in flight, precise analysis of the respiratory gases and oxygen consumption measurements become more difficult. The diluter-demand oxygen regulator provides a mixture of oxygen and air to the pilot. The regulator adds a minimum of 0% oxygen to a maximum of 30% oxygen at sea level. The mixture varies according to outlet flow and altitude as shown in Table XI (Ref 22). Because of varying mixtures, the oxygen consumption analysis involves measuring the inspired and expired volumes and flow-volume averaging the inspired and expired oxygen contents (Ref 23). This method allows a single breath analysis of oxygen consumption using IFPDAS measured parameters. The Beckman OM11 oxygen sensors provide measurements of oxygen partial pressure and the flow sensors provide measurements of flow volume. For this application, the OM11 sensors require the lead compensation discussed in Appendix G which decreases the sensor response time to 100 msec. The remaining discussion presents the mathematical anaylsis and the IFPDAS II implementation.

Table XI

Oxygen Ratio

ALTITUDE (1000 FEET)	OUTLET FLOW (LITERS/MINUTE)	% OXYGEN ADDED FROM SOURCE	
		MINIMUM	MAXIMUM
0	15	0	30
0	50	0	30
5	15	1	33
5	50	1	33
10	15	6	45
10	50	6	45
10	135	6	60
15	15	14	52
15	50	14	52
15	135	14	70
20	15	24	55
20	50	24	55
20	135	24	80
25	15	40	80
25	50	40	80
25	135	40	90
28	15, 50,135	60	100
32	135	98	100
	With diluter at 100%		
All altitudes	15, 50, 135	98	100

(Ref 22)

The instantaneous expired oxygen volume is related to the partial pressure of oxygen by

$$\frac{d\,V_{E_{O2}}}{dt} = \frac{d\left\{ V_E \cdot P_{E_{O2}} / P_{Abs} \right\}}{dt} \qquad (22)$$

where,

$V_{E_{O2}}$ = Volume of expired oxygen (liters)

V_E = Volume of expired air (liters)

$P_{E_{O2}}$ = Partial pressure of oxygen expired (mm Hg)

P_{Abs} = Absolute pressure (mm Hg)

Since,

$$P_{E_{O2}} / P_{Abs} = F_{E_{O2}} \tag{23}$$

where $F_{E_{O2}}$ is the oxygen fraction in the expired air.

Using Eq (23), Eq (22) becomes,

$$\frac{d\ V_{E_{O2}}}{dt} = \frac{d \left\{ V_E \cdot F_{E_{O2}} \right\}}{dt} \tag{24}$$

The flow-weighted expired oxygen volume then becomes,

$$V_{E_{O2}} = \int_{\text{breath}} F_{E_{O2}} \cdot V_E \quad dt \tag{25}$$

where $V_{E_{O2}}$ is the expired volume of oxygen in liters/breath

A similar computation for inspired oxygen volume is made and the per breath oxygen consumption can then be computed:

$$V_{O2} \text{ (liters/breath)} \quad = \quad V_{I_{O2}} - V_{E_{O2}} \tag{26}$$

where,

V_{O2} = Volume of oxygen consumed (liters/breath)

$V_{I_{O2}}$ = Inspired volume of oxygen (liters/breath)

The expired oxygen volume measurement is made using the curves of expired flow volume rate and oxygen partial pressure as shown in Figure 33. The per breath expired oxygen volume is the area under the $(F_{E_{O2}})(V_E)$ curve measured over the period of one breath. A trapezoidal approximation to the area under this curve will yield the flow-weighted expired oxygen volume:

$$V_{E_{O2}} = \left[\tfrac{1}{2} \left\{ F_{E_{O2_1}} \cdot \dot{V}_{E_1} \right\} + \sum_{i=2}^{n-1} \left\{ F_{E_{O2_i}} \cdot \dot{V}_{E_i} \right\} + \tfrac{1}{2} \left\{ F_{E_{O2_n}} \cdot \dot{V}_{E_n} \right\} \right] t \tag{27}$$

where,

$F_{E_{O2_i}}$ = Expired oxygen fraction value

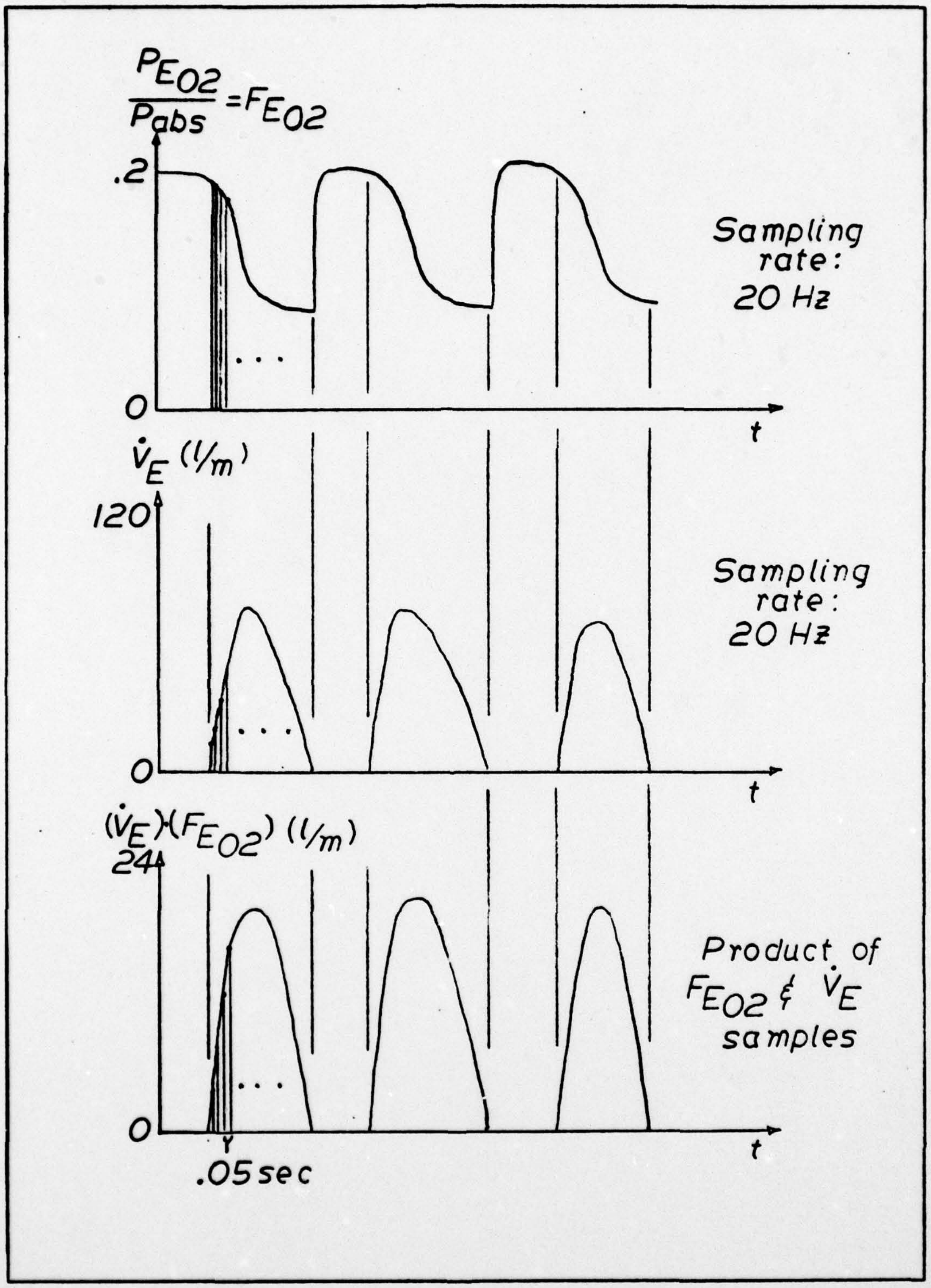


Fig. 33. Flow Rate Weighted Expired Oxygen Measurement

V_{E_i} = Expired volume flow rate sample value (liters/min)

t = Sample to sample width (min)

A similar method is used to determine the inspired oxygen volume and the pilot's oxygen consumption is then computed:

$$V_{02} \text{ (liters/min)} = V_{I_{02}} - V_{E_{02}} \qquad (28)$$

The IFPDAS II operating system can be modified to store all of the parameters required for an accurate calculation of the oxygen consumption if an inspired flow sensor is acquired. The required flow-weighted sums are calculated as follows: As the inhaled breath is detected, the inhaled flow rate/oxygen-fraction product is calculated. These samples are summed over the period of the inspired breath and stored. A similar product-sum is calculated for the period of the exhaled breath and stored. These values are then available for altitude correction and integration on the ground.

A sampling rate of 20 Hz is adequate to insure minimal error for the trapezoidal approximation of the integral as the breathing rate averages only .2 Hz. This routine also allows ample time to perform the necessary multiplication and division routines, each requiring a maximum of 380 microsec. Measurement delays can be compensated for by temporarily storing sample values and time-adjusting them to provide maximum accuracy.

VITA

Joseph Gregory Jolda was born on 12 January 1948 in Worcester, Massachusetts. He graduated from high school in Webster, Massachusetts, in 1965 and attended Northeastern University, Boston, Massachusetts, from which he received the degree of Bachelor of Science in Electrical Engineering in June 1970. Upon graduation he was employed as an associate engineer for Amperex Electronic Corporation, Slatersville, Rhode Island. He also taught Mathematics and Circuit Theory at Worcester Junior College, Worcester, Massachusetts. He then entered active duty in September 1970, and received his USAF commission from OTS in January 1971. He completed pilot training and received his wings in April 1972. He served as a T-33 pilot with the 67th Tactical Reconnaissance Wing at Bergstrom AFB, Texas, and then as a T-37 instructor pilot with the 37th Flying Training Squadron at Columbus AFB, Mississippi, until entering the School of Engineering, Air Force Institute of Technology, in June 1976. He is a member of Eta Kappa Nu and Tau Beta Pi.

Permanent Address: Wawela Park
Webster, Massachusetts 01570

VITA

Stephen John Wanzek was born on 23 September 1948 in Racine, Wisconsin. He graduated from high school in Northbrook, Illinois in 1966 and attended the University of Illinois from which he received the degree of Bachelor of Science in Electrical Engineering and Computer Science in June 1970. After graduation, he was employed as a system programming engineer for Automatic Electric Company, Northlake, Illinois. In November 1970 he entered Officer Training School and was commissioned in February 1971. He completed pilot training and received his wings in February 1972. He then served as a T-37 instructor pilot in the 37th Flying Training Squadron and as a T-37 flight examiner in the 14th Flying Training Wing, Columbus AFB, Mississippi until entering the School of Engineering, Air Force Institute of Technology, in June 1976. He is a member of Eta Kappa Nu and Tau Beta Pi.

Permanent Address: 1551 Chapel Court
Northbrook, Illinois 60062

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
14 AFIT/GE/EE/77-21

2. ~~GOVT ACCESSION NO.~~ 3. ~~RECIPIENT'S CATALOG~~ NUMBER
9 Master's thesis.

4. TITLE (and Subtitle)
6 AIRCREW INFLIGHT PHYSIOLOGICAL DATA ACQUISITION SYSTEM II,

5. ~~TYPE OF REPORT & P~~ERIOD COVERED
MS Thesis

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 Joseph [redacted] Jolda   Gregory   Stephen [redacted] Wanzek   John
[redacted]

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Institute of Technology (AFIT-EN)
Wright-Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK ~~AREA & WORK UNIT NUMBERS~~
12 179p.

11. CONTROLLING OFFICE NAME AND ADDRESS
USAF SAM/VNE
Brooks AFB TX 78235

11

12. REPORT DATE
Decem[redacted]977

13. ~~NUMBER OF PAGES~~
178

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release, IAW AFR 190-17

JERRAL F. GUESS, Captain, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Data Acquisition
Microprocessor Controlled
Bubble Memory
Physiological Measurement
Inflight Physiological Data Acquisition
Data Reduction

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper discusses a second generation microprocessor-based prototype system to acquire, analyze, and store selected environmental and physiological data from a pilot during flight. The Aircrew Inflight Physiological Data Acquisition System (IFPDAS) II consists of an input multiplexer and analog-to-digital converter, a heart rate detector, a microprocessor, and a permanent memory device. The microprocessor's operating system monitors eight sensors, extracts desired information, and stores these reduced data in permanent memory. After the flight,

012 225   Gut

these data are transferred to a land-based computer which completes the data processing and graphs the following environmental and physiological information versus flight time: (1) cabin absolute pressure, (2) cabin altitude, (3) Z-G's, (4) heart rate, (5) breathing rate, (6) minute ventialtion volume, (7) inspired oxygen quantity, and (8) expired oxygen quantity.

The completed IFPDAS II prototype provides the desired information well within the required accuracy. It provides the following parameter ranges: (1) heart rate from 53 ± .1 to 225 ± 2.2 b/min, (2) breathing rate from 4.7 ± .1 to 50 ± 1 b/min, (3) minute ventilation volume from 0 to 100 ± 2 l/min, (4) absolute pressure from 0 to 760 ± 2 mm Hg, and (5) G's from -3 to +12 ± .1 G.