P-WAVE CONTROLLER

FOR A TOTAL ARTIFICIAL HEART

A Thesis

Presented to the

Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

of the Requirement for the Degree

Master of Science in Electrical Engineering

---

by

Bhasker Vajapeyam

May 1978

ACKNOWLEDGMENT

I wish to take this opportunity to express my deep gratitude to Dr. McInnis for his guidance and encouragement during my research in this topic. I would also like to thank Dr. Everett and Dr. Johnson for serving on the committee. I also wish to acknowledge the help received from other graduate students and laboratory personnel. My special thanks are also due to Ms. Michele Maes for typing under severe constraints of time.

P-WAVE CONTROLLER

FOR A TOTAL ARTIFICIAL HEART

An Abstract of a Thesis

Presented to the

Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

of the Requirement for the Degree

Master of Science in Electrical Engineering

---

by

Bhasker Vajapeyam

May 1978

ABSTRACT

This thesis presents research related to the development of an automatic controller which modulates cardiac output of the Atkutsu artificial heart as a function of the frequency of the P-wave. This neural signal is obtained from electrodes implanted on the natural atrium. The P-wave controller, which has been designed and built, detects the successive pulses, measures the time interval between pulses, and determines the timing intervals of diastole and systole for the artificial heart.

Of particular interest is the fact that the P-wave controller is designed around a single board computer. Experiments with different control algorithms can be carried out merely by changing the software.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

On an average one death out of every three is caused by heart disease. Consequently, considerable research is being done on heart transplants and total artificial hearts. Because of the magnitude of the problems encountered in the development of a total artificial heart it appears that at least two more decades will be required before the art of making a permanent replacement heart is mastered. However, one immediate goal is to produce artificial hearts which will keep human beings alive until a suitable natural heart is available for transplant.

Efforts to improve existing artificial hearts are taking place not only in this country but also in the Soviet Union, Germany, Japan, and Czechoslovakia. Scientists in all these countries have three major common problems:

a) Despite great advances in chemistry there is no athrombogenic material which will guarantee that blood will not clot even after contact with it for a long time.

b) Another problem is to find a suitable energy source for the artificial heart. An artificial organ which can work for seventy years must have an independent energy source inside it.

c) The third problem is to control the artificial heart. The natural heart has a very sophisticated intrinsic control system to dynamically adjust the cardiac output to the level needed by the body. The natural heart seems to take into account neural signals, systemic

pressures and several other known and unknown variables to accomplish its task. Control systems currently used for control of artificial hearts are either manual or semi-automatic with a very limited range of cardiac output. The influence of the neutral control system on myocardial activity has been totally neglected.

The purpose of this thesis is to present the design of a P-wave controller, which modulates cardiac output as a function of neural signals. The block diagram of the P-wave controller is shown in Figure 1. It is one section of a more comprehensive control system which is shown in Figure 2.

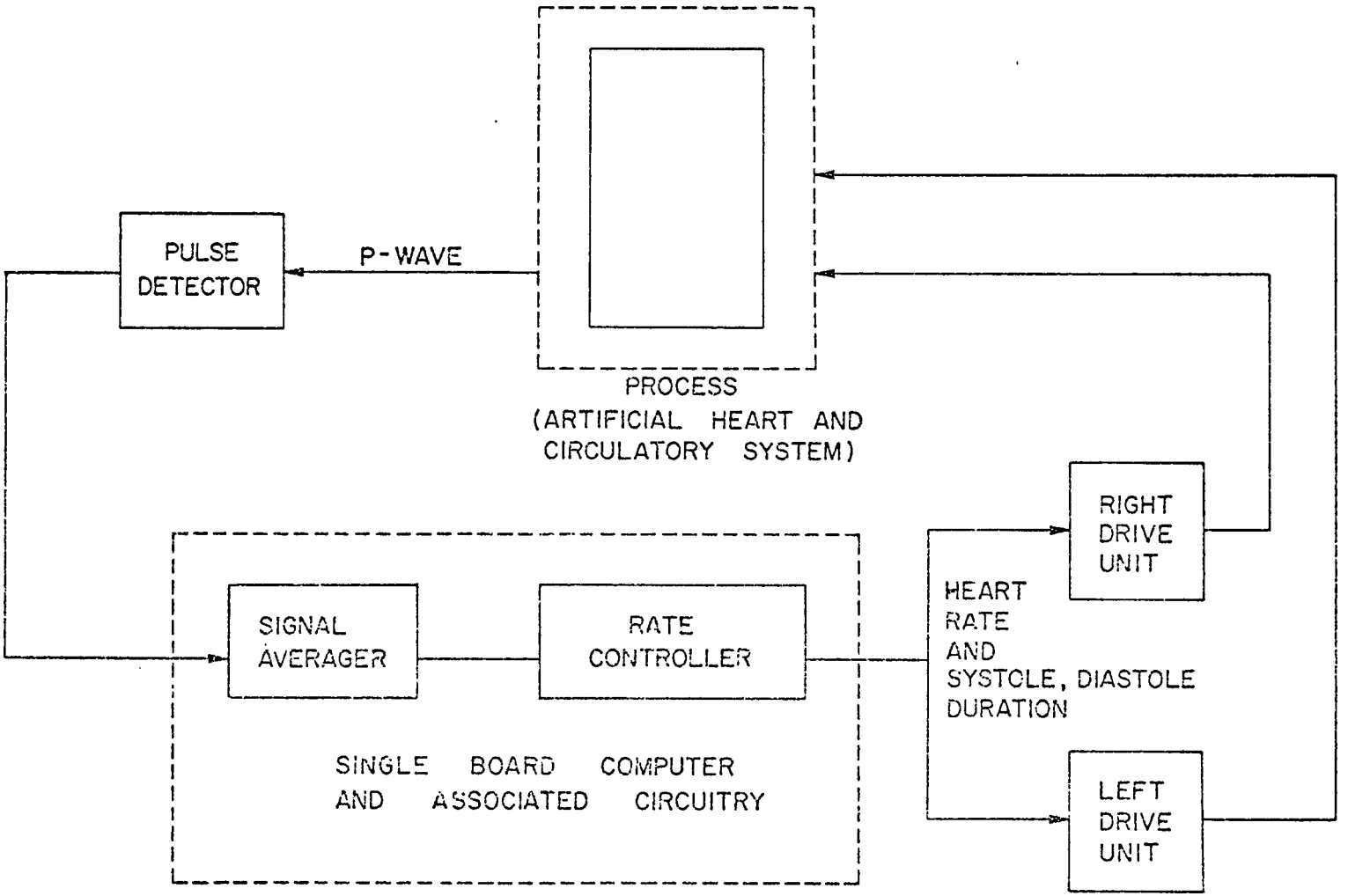


FIG. 1 BLOCK DIAGRAM OF P-WAVE CONTROLLER

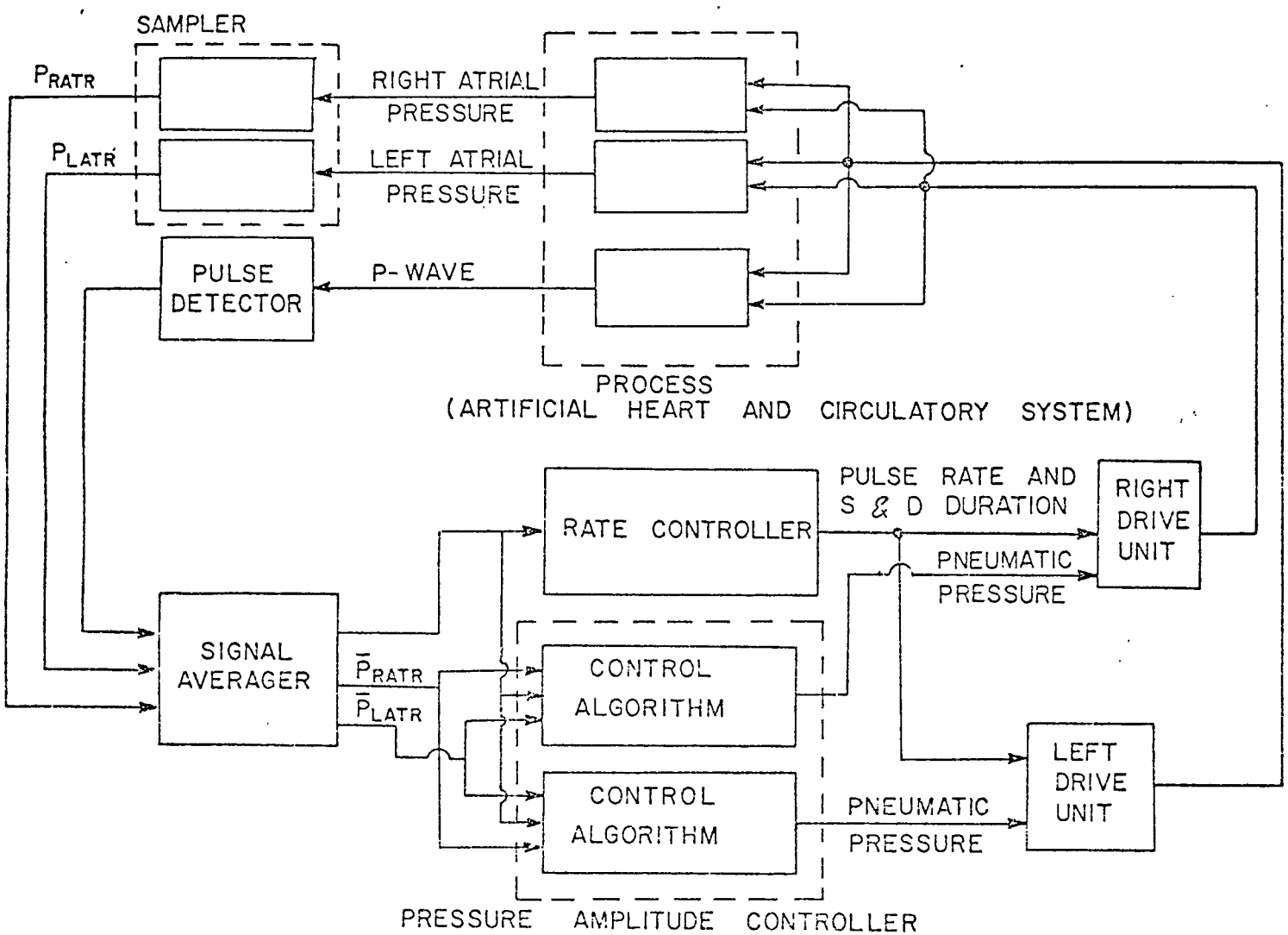


FIG. 2 DIGITAL COTROLLER DIAGRAM

CHAPTER 2

BACKGROUND

Currently, most control systems for artificial hearts consist of a single input/single output feedback system, and their effective range of cardiac output is quite restrictive. Efforts to improve existing and develop new control systems for the total artificial heart are currently taking place at three centres in the United States: Dr. Kolff's group in Salt Lake City, Utah (Kolff, et. al.[1]); Dr. Nose's group in Cleveland, Ohio; and Dr. Pierce's group at Pennsylvania State University (Pierce et. al.[2]). Simultaneously groups in Germany (Newsmann[3]), Japan and Czechoslovakia (Klimes et. al.[4]) are attempting to design automatic control systems for the implanted artificial heart. A survey of results through 1973 is given by Gibson.[5]

The objective of automatic control for the artificial heart is to dynamically adjust cardiac output to the level needed by the body at any given time. Most control designs have focused on reproducing in the circulatory system with the implanted artificial heart the static output versus pressure relations measured in the natural cardiovascular system. The efforts have been directed towards reproduction of the cardiac function curve (i.e. cardiac output versus mean right atrial pressure) (Honda et. al.[7]; Kwan-Gett,[6] et. al.).

Generally, increased cardiac output is achieved by increases in the heart rate and/or stroke volume. Frequently, only the stroke volume is changed to satisfy Starling's Law. Klain, et. al.[9] , have reported on a closed loop feedback system for maintaining desired atrial

pressure. The heart was driven at 80 beats/min. with a fixed vacuum during diastole, while the pressure being applied during systole was modulated as a function of atrial pressure. By increasing the driving pressure during systole, the stroke volume was increased. The atrial pressure was measured directly with a transducer. Kolff and Lawson maintain that multiple in dwelling transducers are incompatible with long survival and report on a non-invasive heart monitor which estimates systolic peak pressure and diastolic filling pressure from the contour of the driving pressure. Recently Pierce, et. al.,[2] derived aorta pressure and left atrial pressure from the pneumatic drive line pressure.

In the nearly twenty years since the first implantation of a total artificial heart by Akutsu and Kolff[8] in a dog, significant progress has been made in many areas pertinent to the longevity of the experimental animals. The length of survival times has steadily increased (Kolff and Lawson[1]). However advances in automatic control of the heart have been less spectacular. Most of the control designs are either manual or semi-automatic with atrial pressure control as the main feedback loop. The response of these systems to changes in cardiac demand has been slow, insufficient or dependent on frequent human monitoring and intervention. The time seems right to initiate a new phase in the control of the total artificial heart in which progress will be marked by increasingly automated drive systems capable of faster and better response to the animals demand for circulating blood. The clue to this progress is likely to be in further exploitation of the dynamic interaction of both neural and pressure signals by means of multivariable control theory. Clearly for the artificial heart efforts must go beyond simulating the effects of Starling's Law.

CHAPTER 3

RATIONALE

In the development of an automatic control system our approach is to imitate the natural control system of the heart. In order to realize the maximum range of cardiac output, a multivariable control system which takes into account the dynamic interactions between the various segments is needed. These dynamic interactions include both neural discharge to the cardiovascular system and the pressure relations throughout the cardiovascular system. Cardiac output seems to be mainly controlled by two important processes:

a) Starling's Law behaviour regulates cardiac output as a function of demand as indicated by venous return.

b) Neural control, mediated by low cerebral structures, modulates cardiac output in response to changes in physiological parameters, such as arterial pressure and gas saturation levels.

In order to vary cardiac output as a function of these two processes research at the University of Houston is directed toward the development of a control system which accepts the following inputs:

a) P-wave

b) Left and right atrial pressures

The control system will dynamically control the heart rate, systolic and diastolic time invervals, ventricular pressures and hence regulate the cardiac output.

With the requirements of the proposed control system in mind, a P-wave controller has been designed. It receives the P-wave from electrodes implanted near the SA node of the natural atrium (when the total artificial heart is implanted the top portion of the natural auricles, including the SA node, is left intact). The output of the P-wave controller controls the heart rate and the systolic and diastolic time intervals.

The P-wave controller has the important features that it is built around an INTEL SBC 80/05 single board computer which has a multibus architecture. It can therefore be easily expanded into a multiprocessor, multivariable system which can accept additional inputs.

CHAPTER 4

METHODS

This chapter describes the operation of the P-wave controller. It controls the timing of the Akutsu artificial heart through a pneumatic drive unit. Let us take a brief look at the Akutsu artificial heart and the pneumatic drive unit before considering the P-wave controller.

The Akutsu Artificial Heart

Figure 3 shows a schematic of one ventricle of the Akutsu artificial heart. It consists of a blood chamber housed in a strong outer covering made of polyurethane. Its contoured base plate is also made of the same material. A flexible diaphram is attached to the edge of the base plate. A pneumatic drive line is connected to an opening in the base plate. The blood chamber housing also has two openings with valves V1 and V2. The construction of the other ventricle is identical.

When a high pressure is applied by the pneumatic drive unit, the flexible diaphram is pushed into its full systole position as shown in Figure 3. This causes valve V2 to close; valve V1 opens and blood is forced out of the chamber. When a low pressure is applied, the diaphram is pulled into its full diastole position as shown by the dotted line in Figure 3. This causes valve V1 to close; valve V2 opens causing blood to be drawn into the chamber through the inlet. Alternate cycles of vacuum and pressure are applied causing the pumping action of the heart.

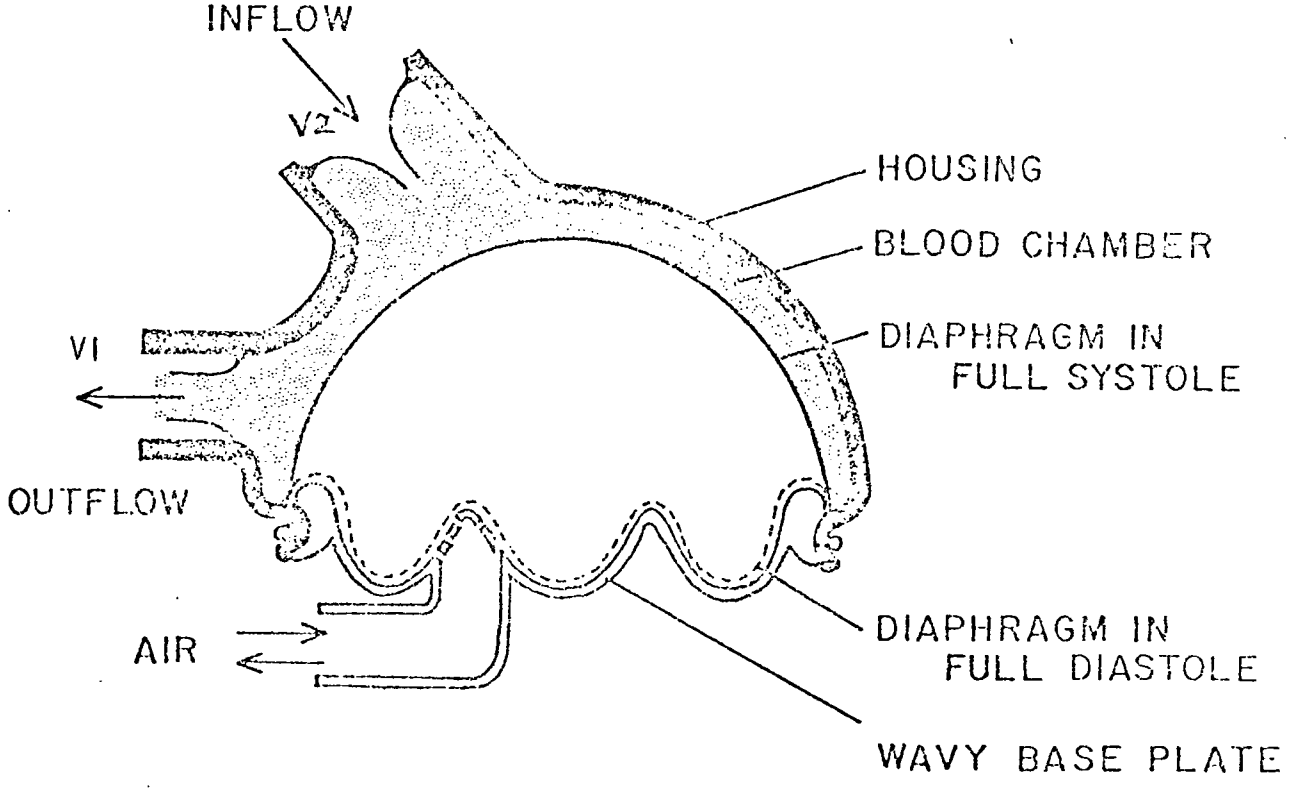


FIG. 3 SCHEMATIC OF ONE VENTRICLE OF THE AKUTSU ARTIFICIAL HEART

The Pneumatic Drive Unit

Figure 4 shows a schematic of the pneumatic unit used to drive the Akutsu artificial heart.

The pneumatic drive unit consists of two sections: the pressure section and the vacuum section. Air from the pressure source is connected to a distributor through an air filter. The distributor has three outlets. One outlet is connected to a pressure relief valve. If the pressure from the source exceeds about 150 psi the safety valve opens and releases the pressure. Air pressure from this point (about 30 psi under normal operating conditions) is used to drive the solenoid valves. The two remaining outputs of the distributor are connected to pressure regulators. The reason for using two pressure regulators is that the pressure levels for the two ventricles are different. The same is true for the vacuum levels. The output of the pressure regulators are connected to solenoid valves through pressure tanks. The other input to the solenoid valves is from the vacuum sections. The operation of the vacuum section is similar except that there is no safety valve. Instead there is a valve that allows the use of atmospheric pressure instead of vacuum from the vacuum source. Alternate cycles of pressure and vacuum are applied to the artificial heart by switching the solenoid valve between pressure and vacuum sections of the unit.

Operation of the P-Wave Controller

In several experiments conducted by Dr. Akutsu at the Texas Heart Institute, electrodes were implanted near the SA node of the natural atrium and the resulting P-wave was recorded on a Hewlett-Packard strip chart recorder. The P-pulse is 90-120 milliseconds wide

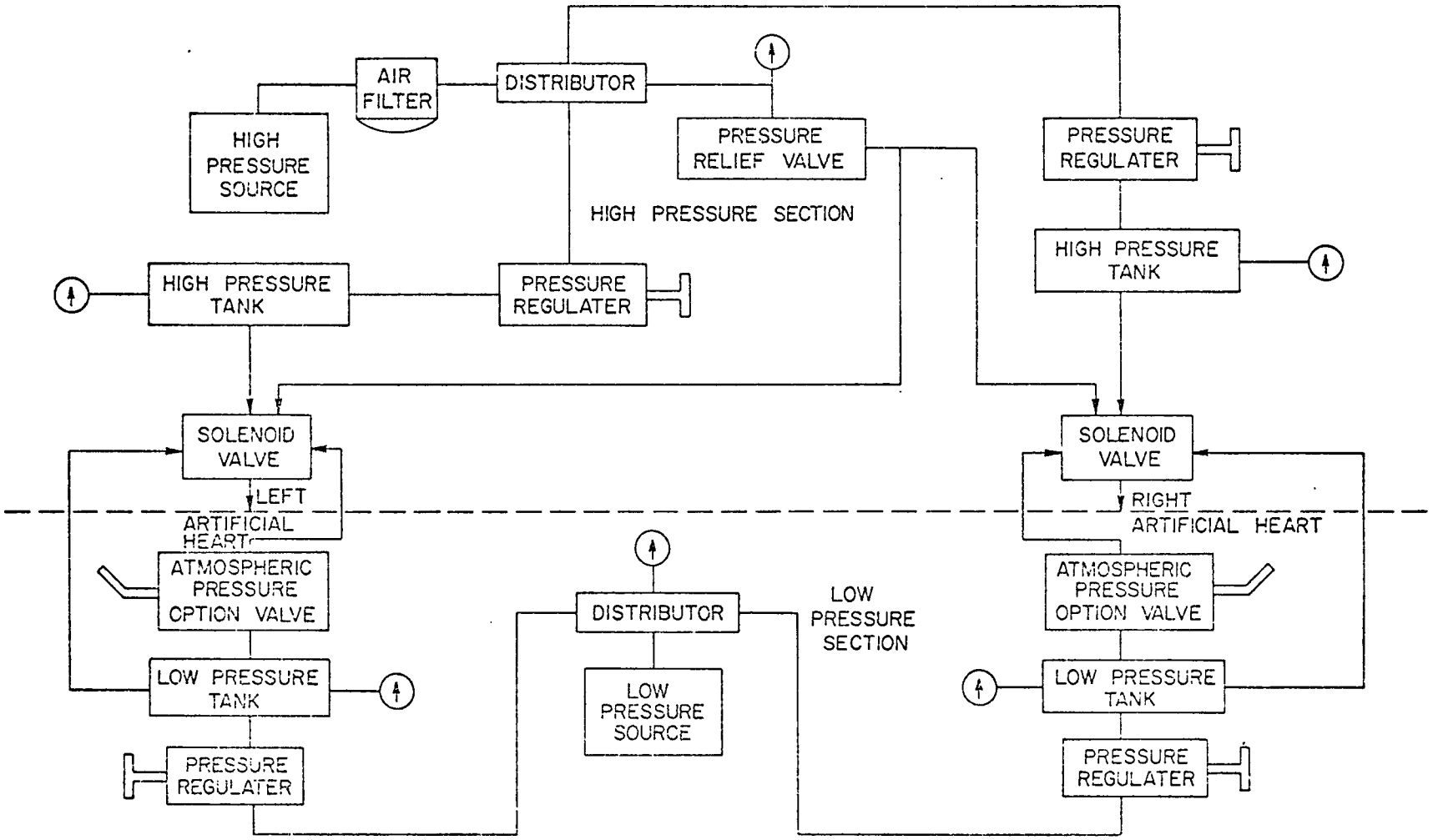


FIG. 4 BLOCK DIAGRAM OF PNEUMATIC DRIVE UNIT

and its amplitude as recorded at the output of the electrodes is about 5 mv. Of course the signal is accompanied by noise.

The first problem therefore is to detect the P-wave in a noisy signal and get a pulse output for every P-pulse input. This task is accomplished by passing the P-wave through a filter circuit which gives a pulse output if two conditions are met:

a) The frequency of the input signal is between 2 Hz and 30 Hz

b) The amplitude of the signal is above 4 mv. (adjustable)

The next problem is to measure the time interval between successive pulses. The systolic and diastolic time intervals are then computed and the beginning of the systole is actuated about 100 milliseconds after the beginning of the P-wave.

The time interval between successive P-pulses is measured in the following manner: A counter is set up to count at some known frequency (in our circuit a frequency of 8 KHz is used). The counter is reinitialized on the positive edge of every P-pulse. The count on the counter just before it is reinitialized is a measure of the time interval between the two preceding P-pulses.

It is important to mention at this stage that in our algorithm the preceding time interval alone is not used to compute the systolic and diastolic time intervals, but, in order to smooth out transient irregularities, the average of the preceding four time intervals is used.

A graph of the time interval between heart beats VS systolic time interval (in effect it is a graph of heart rate vs systolic time inverval) is stored permanently in a portion of the memory used in the

system. The average time interval is then located on the graph, i.e. the operating point is determined and the corresponding systolic time interval is read. The diastolic time interval is then computed by subtracting the systolic time interval from the total time interval. This method of computation is used only if the measured average time interval corresponds to a heart rate of between 80 beats/min. and 120 beats/min. If the P-wave frequency is below 80 pulses/min. the systolic and diastolic timing corresponding to a heart rate of 80 beats/min. is used. This is because the body requires a certain minimum cardiac output and since the stroke volume is limited by the size of the artificial heart, in order to maintain at least the minimum cardiac output, the heart rate cannot be allowed to fall below 80 beats/min. If the P-wave frequency is above 120 pulses/min. the heart rate is maintained at 120 beats/min. This is because if the heart rate goes beyond 120 beats/min. there is marked decrease in the stroke volume, and hence also the cardiac output, owing to the inertia of the system.

Once the systolic and diastolic time intervals are computed the systole is triggered about 100 ms after the rising edge of the P-pulse. This delay is approximately equal to the ventricular filling time.

CHAPTER 5

HARDWARE AND SOFTWARE

Detection Circuit

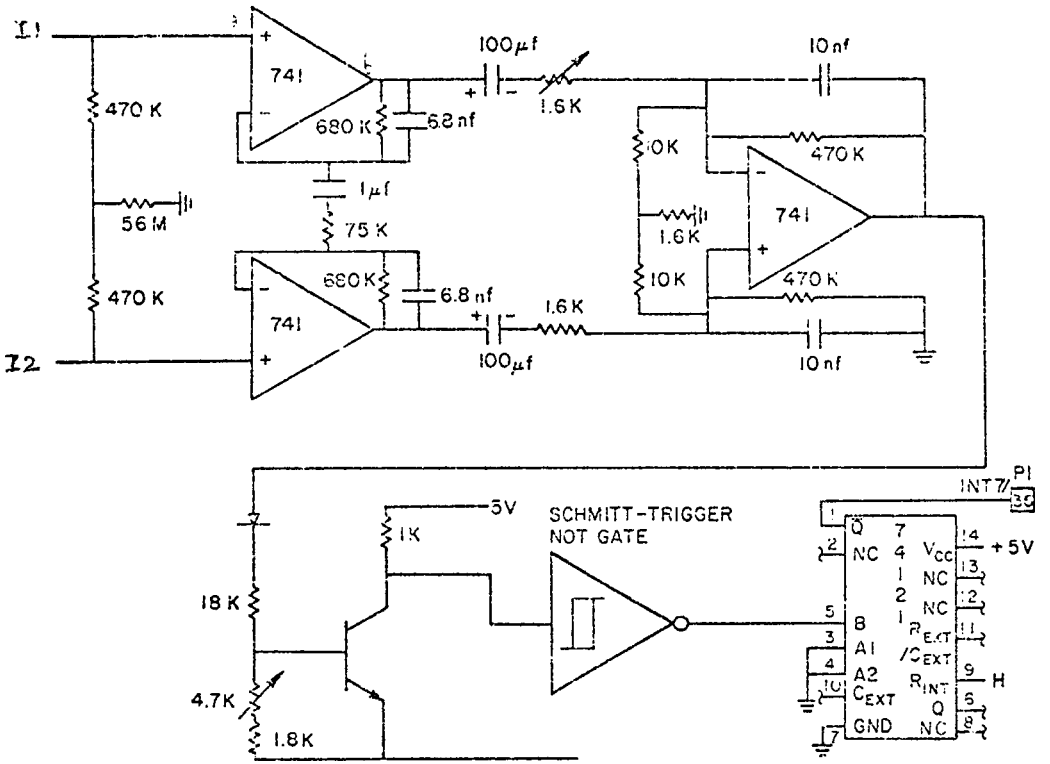
This chapter describes the actual hardware and software design of the P-wave controller. Figure 5 shows the circuit used for detecting the P-wave. The signal from the electrodes is fed in at inputs 11 and 12. The first stage of the detector is a balanced differential amplifier consisting of two μ741 operational amplifiers. The gain of this stage is about 9. It has a low frequency cutoff of 2 Hz and a high frequency cutoff of 34 Hz. The output of this stage is ac coupled to another differential amplifier stage consisting of just one μ741 operational amplifier. The gain of this stage is 293. It has a low frequency cutoff of 2 Hz and a high frequency cutoff of 33 Hz. The output of the second stage is fed into a single transistor circuit which serves two purposes:

a) The transistor is not turned on until the signal voltage at 11, 12 exceeds about 4 mv (adjustable from 2 to 5 mv).

b) The output voltage range of this stage is 0-5 V. It is thus directly connected to a Schmitt trigger NOT gate and thence to a monostable multivibrator which gives a 35 ns pulse for every P-pulse input.

Single Board Computer and Associated Circuitry

The output pulse of the monostable multivibrator is used to interrupt the SBC 80/05 single board computer. The operation of the



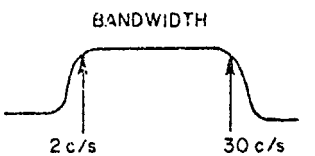


FIG. 5 PULSE DETECTOR

single board computer itself is not described here (see SBC 80/05 User's Manual).

Figure 6d shows the circuit diagram used in conjunction with the single board computer. It consists of a clock generator and counters to generate a frequency of 8 KHz. This frequency is used for clock inputs of the programmable interval timer. The table given below shows the addresses of the different ports of the 8253.

ADDRESS	FUNCTION
08	Counter 0
09	Counter 1
0A	Counter 2
0B	Mode Control Port

Note: Read operation on port 0B is illegal.

The IORC/and IOWC/ are NANDed to produce IORC + IOWC. This signal is further NANDed with CS/ to produce XACK/. In other words if the 8253 is selected and if I/O read or write signal is active an acknowledge signal is sent to the processor, pulling it out of the wait state. The first counter on the 8253 is programmed to operate mode 0 while the other two counters are programmed to operate in mode 1. It may be noted at this stage that counter 1 can be triggered by a low to high transition of the output of counter 2 or bit PC3 of the 8155. Counter 2 can however, be triggered only by a low to high transition of the output of counter 1. Counter 0 on the 8253 interval timer is used to measure the time interval between successive pulses. Every time there is an interrupt from the P-wave the count on this counter is

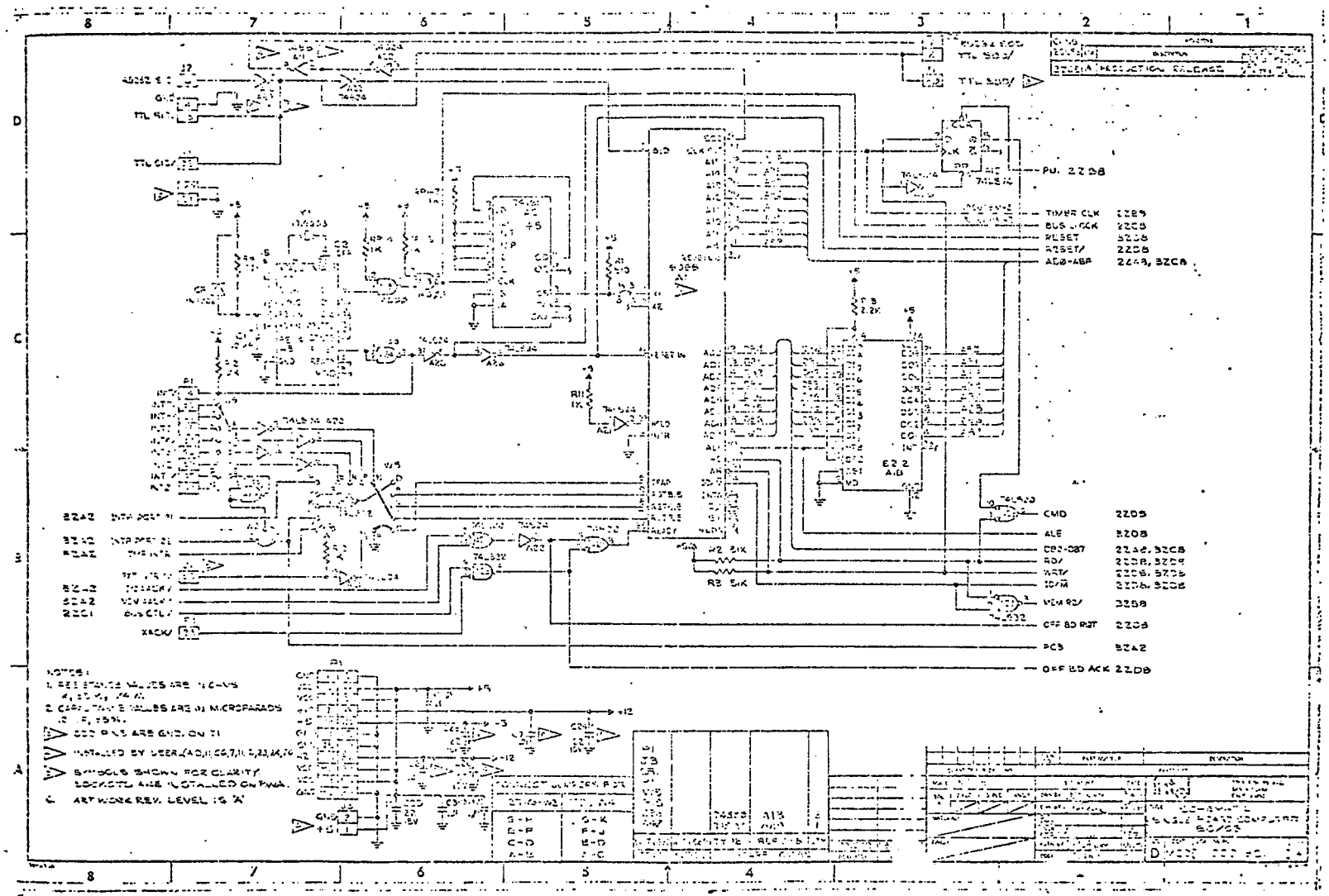


FIG. 6a CENTRAL PROCESSING SECTION OF THE SINGLE BOARD COMPUTER

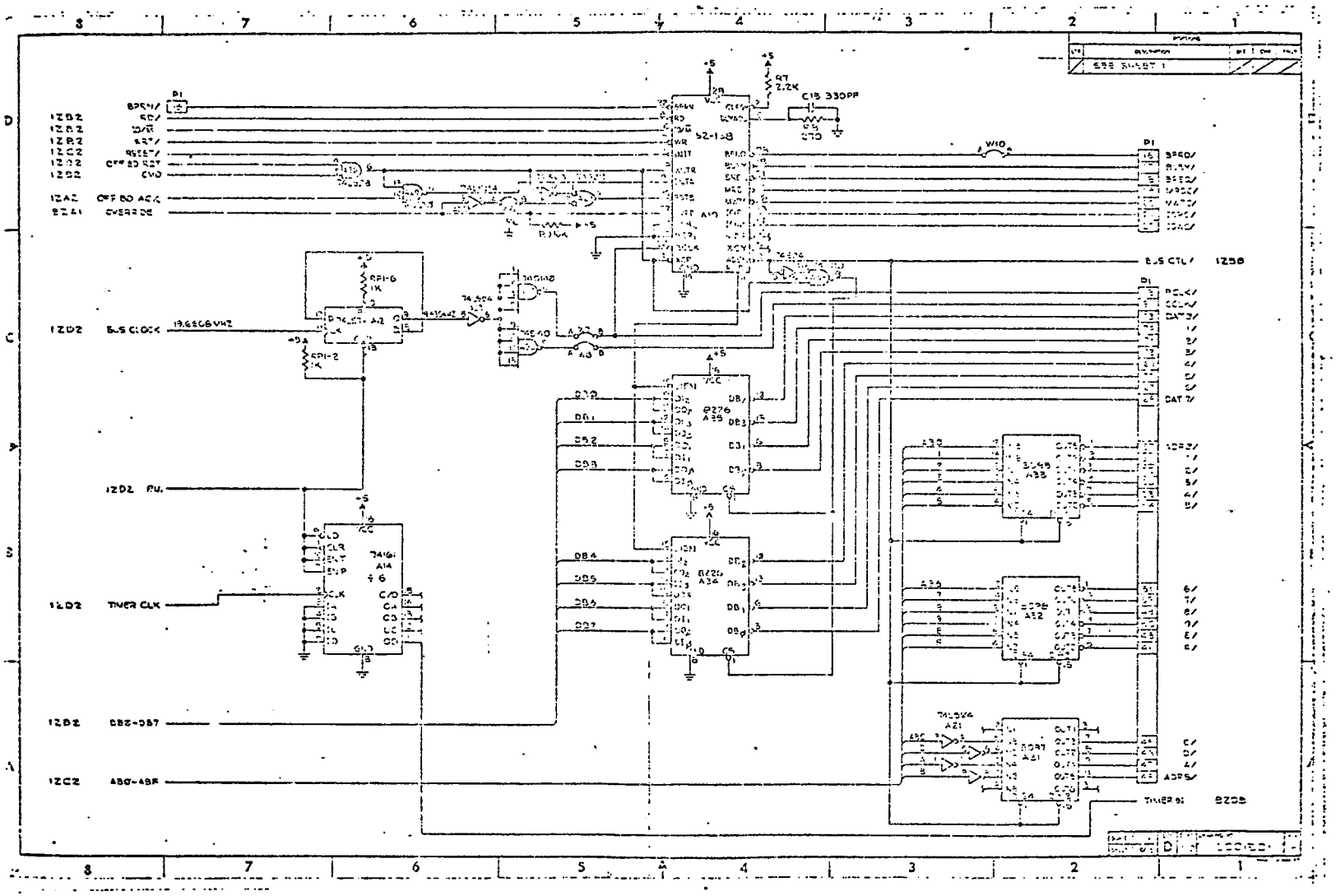


FIG. 6b BUS INTERFACE OF THE SINGLE BOARD COMPUTER

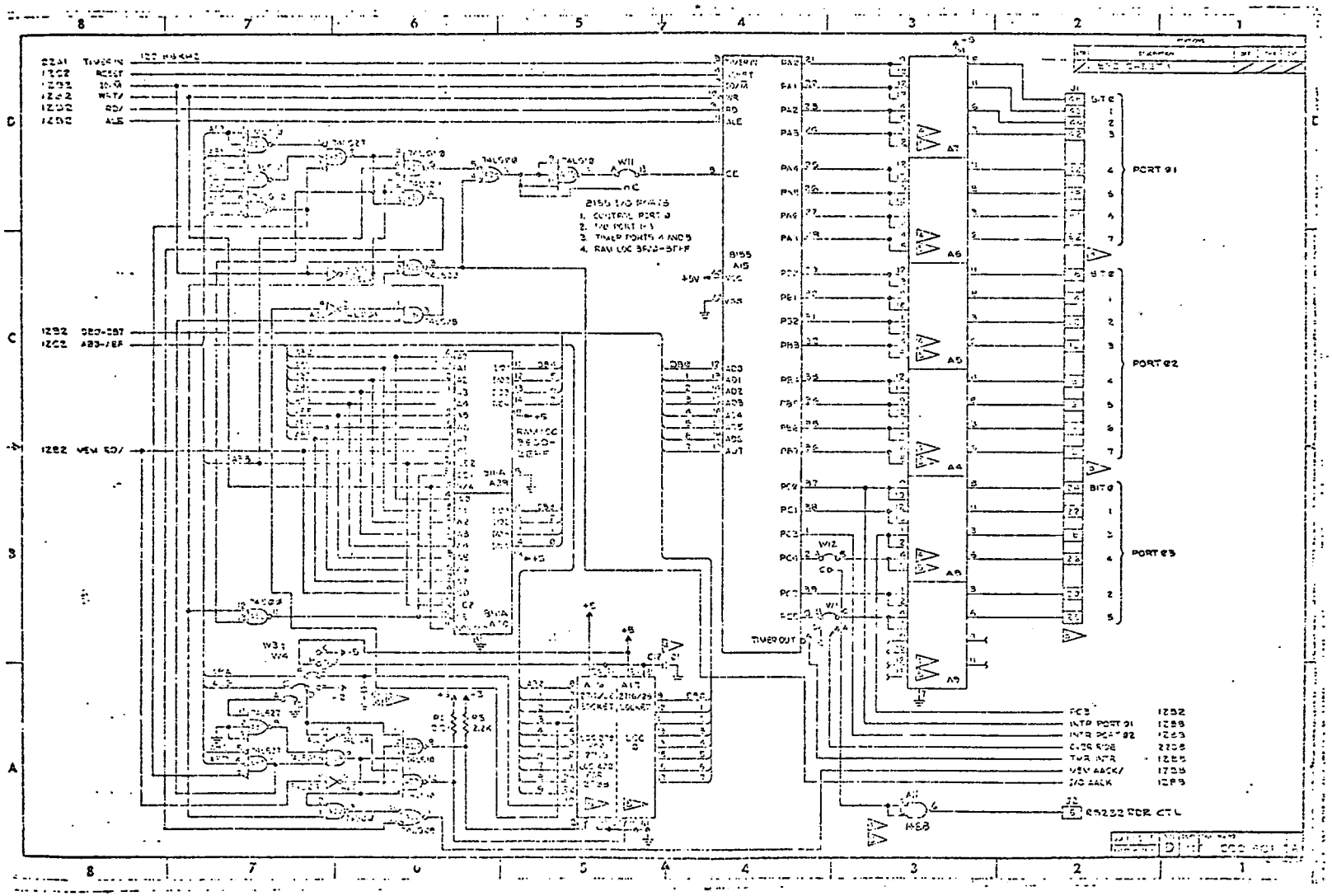
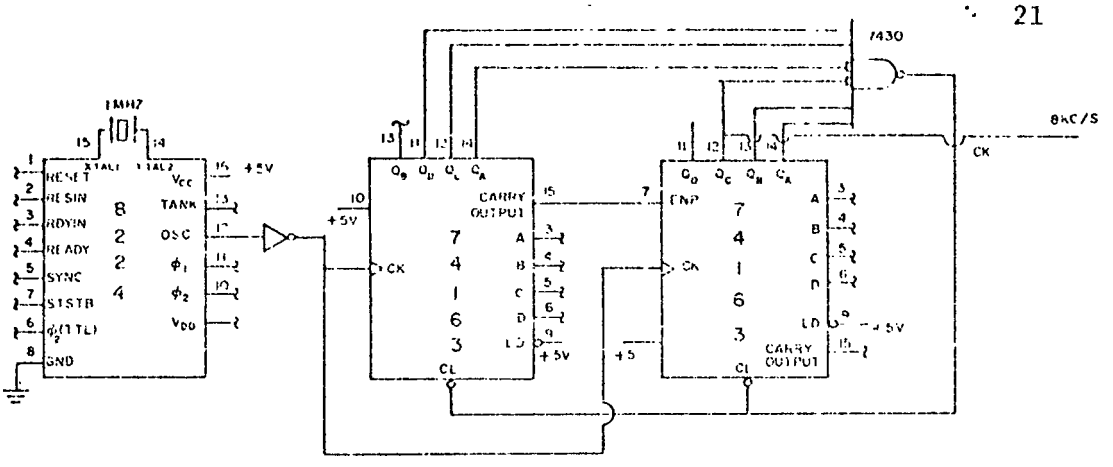


FIG. 6c INPUT OUTPUT AND MEMORY SECTION OF THE SINGLE BOARD COMPUTER



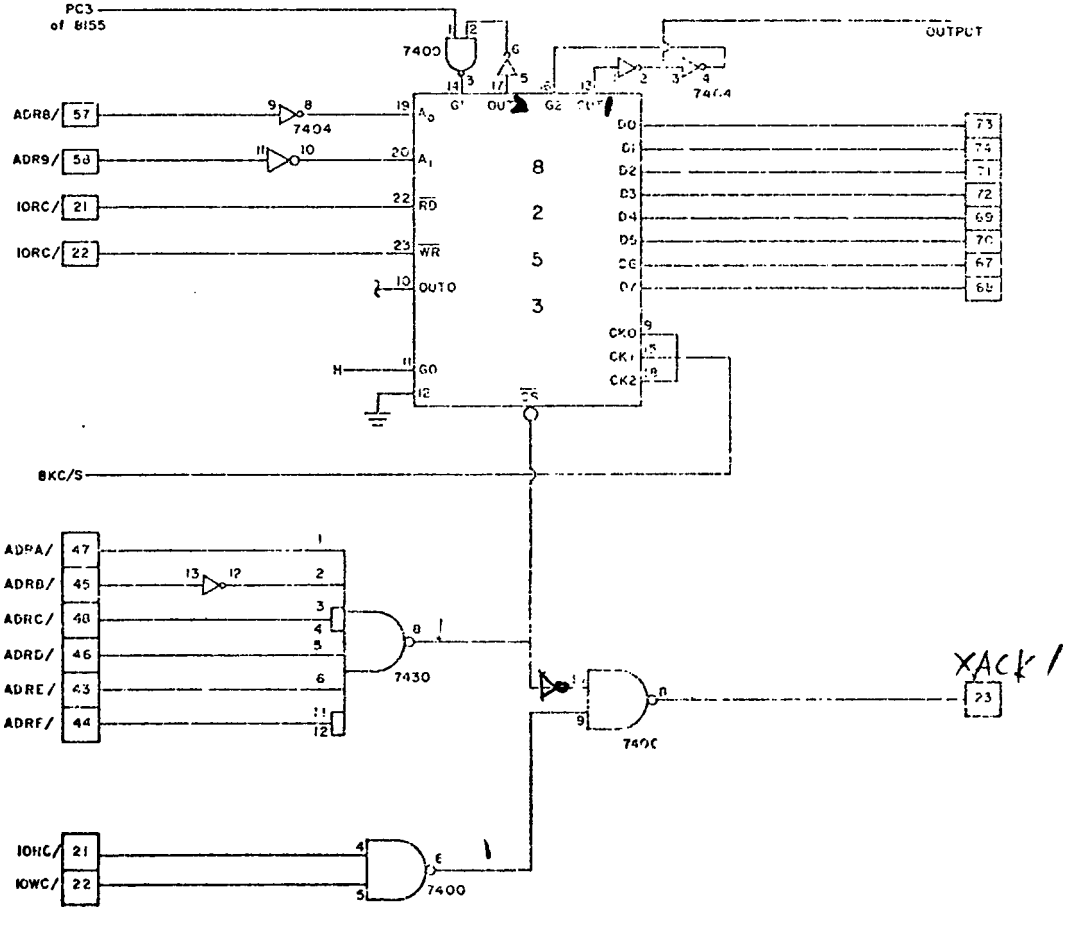


FIG. 6d

FIG. 6d PROGRAMMABLE INTERVAL TIMER AND ASSOCIATED CIRCUITRY

read and reinitialized. The value that is read before reinitialization is a measure of the interval between the preceding two pulses. Counter 1 and Counter 2 are programmed to operate in mode 1. The computer outputs the systolic and diastolic time intervals to these two counters. The output 01 remains high for a period of time equal to the systolic time interval, and remains low for a period of time equal to the diastolic time interval.

To understand how this happens let us consider the operation of the system under two conditions:

a) When the frequency of the P-wave is between 80 pulses/min. and 120 pulses/min.

b) When the frequency of the P-wave is outside the above range.

If the frequency the P-wave is between 80 pulses/min. and 120 pulses/min. the computer calculates the systolic and diastolic time intervals when it is interrupted by the P-wave and goes into a loop to generate a delay of about 15 ms. After this delay is over the systolic time interval is loaded into the counter 1, the diastolic time interval is loaded into the counter 2 and the beginning of the systole is actuated through bit PC3 of Port C of the 8155. Though the beginning of the systole is actuated by the computer about 15 ms after the positive edge of the P-pulse the actual systole starts about 85 ms later. This is due to the delay introduced by the solenoid valve. The output 01 remains high for a period equal to the systolic time interval and then goes low thus actuating the diastole. The systole is retriggered as before after the arrival of the next P-pulse. If, however, the P-wave

suddently fails for any reason the systole is retriggered by the output of counter 2.

If the frequency of the P-wave is below 80 pulses/min. or above 120 pulses/min. the computer just outputs the systolic and diastolic time interval but does not actuate the systole after the arrival of the P-pulse. In other words no effort is made to start the systole 100 milliseconds after the arrival of the P-pulse. The systole is retriggered by the output of counter 2 when the diastolic time interval expires. The diastole is retriggered by the output of counter 1 after the systole time interval expires.

Software

Software for the P-wave controller is divided into eight logical segments:

(1) Initialization

(2) Acquiring data and checking conditions

(3) Adding and averaging

(4) Determining the operating region on the graph

(5) Computing the address of tables

(6) Locating the exact operating point

(7) Computing systolic and diastolic time intervals

(8) Output after delay

(1) Initialization

When the control unit is turned on a single pulse resets the entire system the program counter of the microprocessor is set to 00H and it immediately starts operation. The first few instructions should therefore be an initialization program. All segments of the software

are shown in the Appendix.

As explained earlier in Chapter 4 the average of four previous P-wave intervals is taken into consideration to compute the systolic and diastolic time intervals.This requires eight bytes of RAM (two bytes per interval) to be reserved for storing the real time incoming data. Locations 3E04H through 3E0BH are used for this purpose. Locations 3E02H and 3E03H contain a pointer to the location in which the next incoming interval is to be stored. For example, if the previous P-wave interval was stored in locations 3E06H and 3E07H. Location 3E02 and 3E03 will contain 08H and 3EH respectively. Location 3E00 is initially set tc 04. It is decremented by one very time a P-pulse occurs. When the value in location 3E00 reaches the zero the address pointer is reset to 3E04. Location 3E00 is also reset to 04 at this time.

The initialization routine programs counter 0 of the 8253 to operate in mode 0 and counting is initiated at once. Counter 1 and counter 2 are programmed to operate in mode 1.

The P-wave controller initially outputs a 335 milliseconds systolic time interval and a 415 milliseconds diastolic time interval. These time intervals are arbitrary and correspond to a heart rate of 80 beats/min. The P-wave controller goes into dynamic operation only after the first 5 pulses have elapsed. This is necessary because the real time program has to be provided with sufficient data to carry out the task of averaging and computing of systolic and diastolic time intervals. Location 3E01H is accordingly initialized to zero and incremented by one every time a P-pulse occurs. This is done until the value in the 3E01H reaches five after which the location is used in the real time routine

only to check if five pulses have elapsed.

The stack pointer is next initialized so that every time there is an interrupt from the P-wave the PSW and registers can be pushed on to the stack.

(2) Acquiring Data and Checking Conditions

This segment of the program comes into play as soon as there is an interrupt from the P-wave. The PSW and the processor registers are pushed on to the stack, the H, L register pair is then loaded with the address of the location in which the lower byte of the interval is to be stored. As described earlier, this address is contained in locations 3E02 and 3E03H. The two bytes are then input from the counter 0 and stored in two consecutive locations. The pointer is then reinitialized to 3E04 if the count in location 3E00 is zero. The system then proceeds to compute systolic and diastolic time intervals if the count in location 3E01H is at least five.

(3) Adding and Averaging

This segment of the program is concerned with the adding and averaging of the last four P-wave intervals. As is evident from the flow chart and the program the four values each in two bytes are added together and the result is stored in 3 bytes - 3E0C, 3E0D, 3E0EH. The three bytes are then shifted right twice, this is equivalent to a division by four. The error ± 1 bit equivalent to .016 ms is negligible.

(4) Determining the Region on the Graph

Figure 7 shows a graph of heart rate vs systole, it is divided into 10 regions (0 through 9). In order to determine the operating

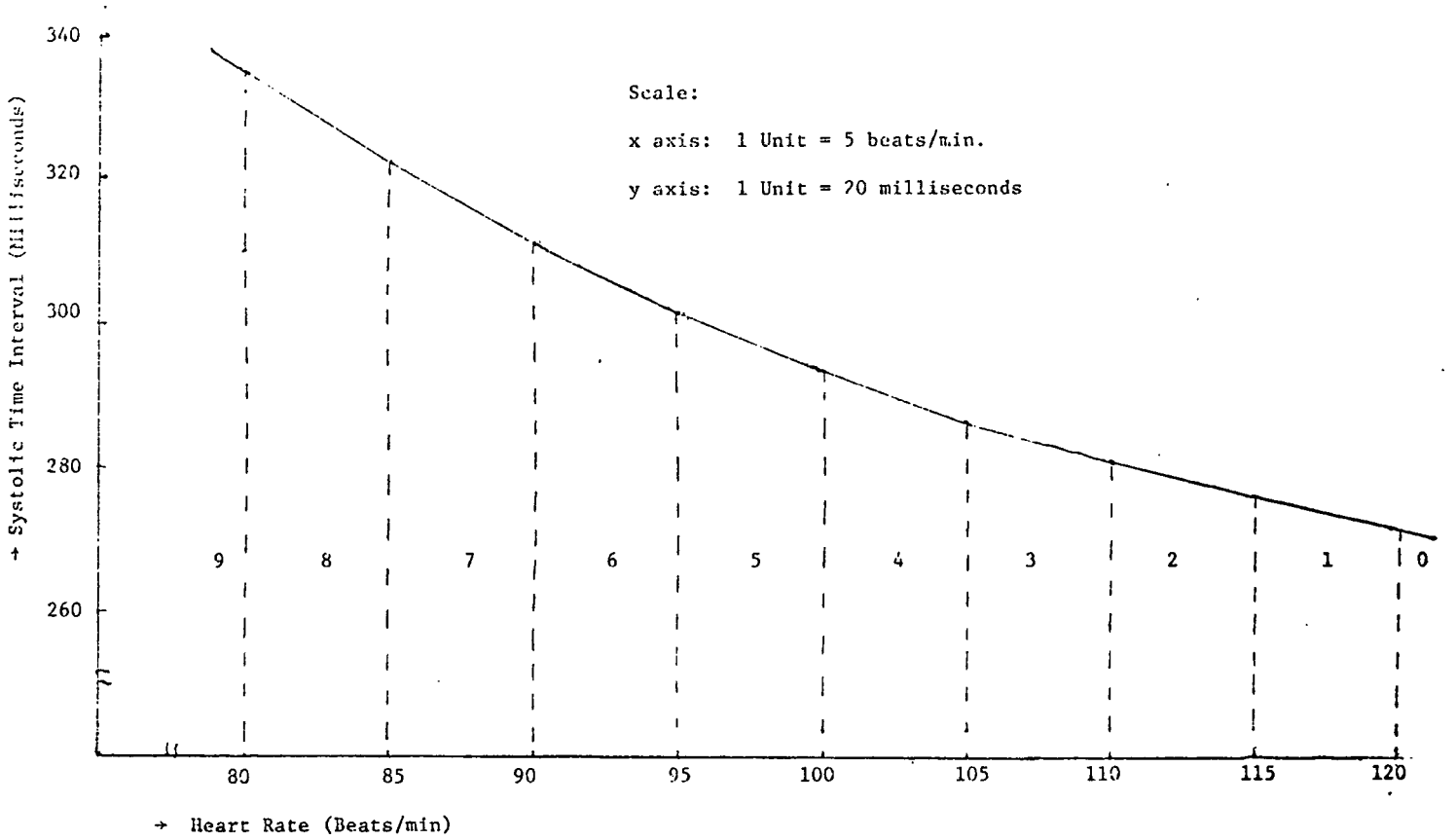


FIG. 7 GRAPH OF HEART RATE VS SYSTOLIC TIME INTERVAL

point the program first determines the region in which the average time interval falls before locating the exact point. This segment of the program determines the operating region on the graph. To do this the most significant bytes of the break points (17,16,14,13,12,11,11,10,0F) are stored in locations 700 through 707H and the least significant bytes (70, 10, D8, C0, C0, D8, 08, 51, A0) are stored in locations 708 through 70F. The average time interval is stored in locations 3E0C and 3E0D. When execution of this segment is finished the region number is stored in 3E0FH. If the system enters operating region 9, the systolic time interval and diastolic timing interval corresponding to a heart rate of 80 beats/min. is stored in locations 3E13 through 3E16 and the output routine is executed. If the system is already in region 9 no action is taken. Exactly the same approach is followed if the system is operating in region zero.

(5) Computing the Address of Tables

In this segment of the program the address of a pointer to a table is computed depending on the value stored in location 3E0F. The operating region was determined in the previous segment of the program. The graph between any two consecutive break points is approximated to be a straight line. In order to find the exact operating point certain parameters such as the slope of the straight line in the particular region must first be determined. This segment chooses the parameters from a table depending upon the operating region and passes them to the next segment which locates the exact operating point.

An example of calculating the parameters (stored in EPROM) of region 8 follows:

If the heart rate is 80 the systolic time interval is = 335 ms

If the heart rate is 85 the systolic time interval is = 322 ms

Therefore the number of discrete points in region 8 = 335-322 = 13

Total time interval for a heart rate of 80 = 750 ms

Count on the counter corresponding to 750 ms = 750 × 8 = 6000 = 1770H

Total time interval for a heart rate of 85 = 706 ms

Count on the counter corresponding to 706 ms = 706 × 8 = 5648 = 1610H

The difference between the two count values (6000 - 5648) is divided by 13 to give the increment in count for 1 ms change in the systolic time interval in this particular region.

i.e. increment in count for

1 ms increase in systolic time interval = $\frac{6000 - 5648}{13}$ = 27 = 1 BH

Location 73A contains this value 1B H. Location 73B contains ODH (13 decimal) the number of increments in region 8. Locations 73CH and 73DH contain the starting point of region 8 (1610H) and locations 73E and 73F contain the starting point of systolic time interval in region 8 (OA10 H). Similar calculations are made for regions one through seven of the graph and stored in locations beginning at 710H. The entire table is given below.

710	26	04	AO	OF	80	08
716	25	05	51	10	AO	08
71C	2A	05	08	11	C8	08
722	1D	08	D8	11	FO	08
728	25	07	CO	12	30	09
71E	1C	OA	CO	13	68	09
734	1C	OB	D8	14	B8	09

(6) Locating the Exact Operating Point

When the system enters this segment of the program locations 3E11H and 3E12H contain the lower unit of the operating region, the increment is in register E and maximum number of increments on register C.

The increment is added to the value in locations 3E11, 3E12H until its value equals or exceeds measured average time interval stored in locations 3E06, 3E0DH. The number of times the increment is added is stored in register B. Since the frequency of the clock is 8 kc/s the value in register B is multiplied by eight and added to the lower limit of the systolic time interval for the particular region stored in location 3E13, 3E14H. This gives the value of the systolic time inverval corresponding to the exact operating point.

CHAPTER 6

DISCUSSION

The development of a rate controller which uses natural neural signals as an input is a major step. The significance of incorporating neural signals into the control system for the artificial heart is yet to be demonstrated. Testing of the P-wave controller with experimental animals at the Texas Heart Institute will provide not only information for improving the controller but information which will be significant in increasing our understanding of the neural control of the natural heart.

BIBLIOGRAPHY

[1] Kolff, W. J. and Lawson, J.: Status of the Artificial Heart and Cardiac Assist Devices in the United States. Trans. Amer. Soc. Artif. Intern. Organs. 21:620, 1975.

[2] Pierce, W. S., Landis, D., O'Bannon, W., Donachy, J. H., White, W., Phillips, W., and Brighton, J. A.: Automatic Control of the Artificial Heart. Trans. Amer. Soc. Artif. Intern. Organs. 22:347, 1976.

[3] Nemsmann, U.: Driving System for Artificial Blood Pumps. Trans. Amer. Soc. Artif. Intern. Organs. 22:128, 1976.

[4] Klimes, F., Skorkovsky, J., Ziga, J., Andrrevosk, F., Bednarik, B. and Vasku, J.: Analysis and Model of Controlling System to Control Heart Rate and Stroke Volumes of an Artificial Heart. Medical Biol. Eng. 13:662, 1975.

[5] Gibson, W. H.: Control of Total Artificial Heart. Artificial Heart Laboratory, Dept. of Surgery, University of Mississippi Medical Center, Jackson, Miss., 1974.

[6] Kwan-Gett, C. S., Crosby, M. J., Schoenberg, A., Jacobsen, S. C. and Kolff, W. J.: Control Systems for Artificial Hearts. Trans. Amer. Soc. Artif. Intern. Organs. 14:284, 1968.

[7] Honda, T., Nagai, I., Nitta, S., Igo, S. R., Edmonds, C. H., Hibbs, C. W., Kito, Y., Fuqua, J. M. and Akutsu, T.: Evaluation of Cardiac Function and Venous Return Curves in Awake, Unanesthetized Calves with an Implanted Total Artificial Heart. Trans. Amer. Soc. Artif. Intern. Organs. 21:362, 1975.

[8] Akutsu, T. and Kolff, W. J.: Permanent Substitutes for Valves and Hearts. Trans. Amer. Soc. Artif. Intern. Organs. 4:230, 1958.

[9] Klain, M., Mrava, G. L., Tajima, K., Schriber, K., Webb, J., Ogawa, H., Opplt, J. and Nose, Y.: Can we Achieve 100 Hours' Survival with a Total Mechanical Heart. Trans. Amer. Soc. Artif. Intern. Organs. 17:437, 1971.

APPENDIX

INITIALIZATION

```
0000    LXI  H, 3E04
        SHLD    3E02        · Initialize pointer
        MVI     A, 04
        STA     3E00
        XRA     A
        STA     3E01
        LXI     SP,3F00     · Initialize Stack Pointer
        MVI     A, 46       · Mode control word of 8155
        OUT     <00>
        MVI     A, 03       · Initially HR is 80
        STA     3E10        · (until 5 P-Pulses elapse)
        MVI     A, 08       · Set PC3 of Port C High
        OUT      03
        JMP     0600        · Continue Initialization Routing
                              at Location 600
```

ACQUIRING DATA AND CHECKING CONDITIONS

```
003C    PUSH    PSW         · Push all registers
        PUSH    B           · on stack
        PUSH    D
        PUSH    H
        EI
        LHLD    3E02        · Load Pointer in H, L
        MVI     A, 00
        OUT     <OB>
        IN      <08>        · Read Count on Counter 0
        MOV      M,A
        INX      H
        IN      <08>
        MOV      M,A
```

```
        MVI     A,30        · Reinitialize Counter 0
        OUT     <OB>
        MVI     A,FF
        OUT      08
        OUT      08
        INX     H
        LDA     3E00        · If four pulses have elapsed
        DCR     A           · reinitialize pointer and location
        JNZ     0066        · 3E0A
        LXI     H,3E04
        MVI     A, 4
0066    STA     3E00
        SHLD    3E02
        LDA     3E01        · If five pulses have not elapsed
        CPI     05          · return otherwise go to 007E
        JZ      007E
        JNC     007E
        INR     A
        STA     3E01
        JMP     01ED
```

ADDING AND AVERAGING

```
007E    XRA     A           · Initialize 3E0C, 3E0D
        STA     3E0C        · 3E0E to zero
        STA     3E0D
        STA     3E0E
        MVI     D,4         · Initialize Reg D to add 4 numbers
        LXI     H, 3E04     · Initialize HL ADDR 3E04
008D    LXI     B, 3E0C     · B, C. Address 3E0C
        MVI     E,2
        XRA     A           · Clear carry bit
0093    LDAX    B
        ADC     M
        STAX    B
```

```
        DCR     E               · Done if E = 0
        JZ      009F
        INX     B               · Point to higher byte
        INX     H               · Point to higher byte
        JMP     0093
009F    INX     B
        LDAX    B               · Load contents of 3E0E
        ACI     0H              · Add carry
        STAX    B
        INX     H
        DCR     D               · If 4 numbers not
        JNZ     008D            · added go to 008D
00A9    XRA     A
        MVI     D,2             · D=2, the no. of shifts to be done
00AC    LXI     B, 3E0E         · B, C = 3E0E
        MVI     E,3             · E=3, no. of bytes to be shifted
00B1    LDAX    B               · Load byte
        RAR                     · Rotate through carry
        STAX    B               · Store it back
        DCR     E
        JZ      00BC            · If 3 bytes rotated go to 00BC
        DCX     B
        JMP     00B1
00BC    DCR     D               · If No. of shifts not
        JNZ     00AC            · equal to 2 go to 00AC
```

CHECKING OPERATING REGION ON GRAPH

```
00C0    LXI     H, 700          · HL = Addr of MSB of Breakpoints
        LDA     3E0D
        MVI     C, 9
        LXI     D, 708          · DE = addr. of LSB of Breakpoints
00CB    CMP     M               · Compare MSBs
        JZ      00DB            · If equal go to compare LSBs
        JNC     00F3            · If MSB of Average Interval is greater
```

```
        DCR     C
        JZ      00F3        · If Reg C = 0 go to 00F3
        INX     H           · If MSB of breakpoint is
        INX     D           · greater than MSB of Average Interval
        JMP     00CB        · compare next breakpoint
00DB    XCHG                · If MSB of Average Interval equals
        LDA     3E0C        · breakpoint compare corresponding
        CMP     M           · LSBs
        JNC     00F3
        JZ      00F3
        DCR     C
        JZ      00F3
        INX     H
        INX     D
        XCHG
        LDA     3E0D
        JMP     00CB
00F3    MOV     A, C
        STA     3E0F
        CPI     00H         · If Frequency of P-wave is greater
        JNZ     0118        · than 120/min and if 3E10 not equal
        LDA     3E10        · to one, output systole and diastole
        CPI     01H         · corresponding to 120 beats/min. store
        JZ      01ED        · 01 in loc 3E10 and return
        LXI     H, 880
        SHLD    3E13
        LXI     H, 720
        SHLD    3E15
        MVI     A, 01
        STA     3E10
        JMP     01CB
0118    CPI     09          · If Frequency of P-wave is less than
        JNZ     0139        · 80 pulses/min. and if 3E10 not equal to 3
        LDA     3E10        · output systole and diastole
        CPI     03          · corresponding to 80 pulses/min. store 03
        JZ      01ED        · in 3E10 and return
```

```
        LXI     H,A78
        SHLD    3E13
        LXI     H, CF8
        SHLD    3E15
        MVI     A,03
        STA     3E10
        JMP     01CB
```

COMPUTE ADDRESS FOR TABLES

```
0139    LXI     H,710           · Depending on count in 3E0F
        LDA     3E0F H          · compute address pointer into table
        LXI     D,0006
0142    DCR     A
        JZ      014A
        DAD     D
        JMP     0142
014A    MOV     E,M             · Reg E = Increment
        INX     H               · in count/ms increment in systole
        MOV     C,M             · Reg C = No. of discrete
        INX     H               · Intervals in the region
        MOV     A,M             · 3E11, 3E12 = Lower
        STA     3E11            · breakpoint of region
        INX     H
        MOV     A,M
        STA     3E12
        INX     H
        MOV     A,M
        STA     3E13            · 3E13, 3E14 = systole
        INX     H               · at lower breakpoint
        MOV     A,M
        STA     3E14
```

ZERO IN ON EXACT OPERATING POINT

```
        MVI     B,0
```

```
        LDA     3E12            · go to 00F3
0166    LXI     H, 3E0D         · Compare MSB of Average Interval
        CMP     M               · with 3E12
        STA     3E12
        LDA     3E11
        JZ      018C            · If equal compare LSBs
        JNC     0194
0176    ADD     E               · If average interval greater than
        JNC     017E            · the no. in 3E11, 3E13 and increment
        LXI     H, 3E12
        INR     M
017E    INR     B
        DCR     C
        JZ      0194
        STA     3E11
        LDA     3E12
        JMP     0166
018C    DCX     H               · Compare LSBs
        CMP     M
        JZ      0194
        JC      0176
```

COMPUTE SYSTOLE AND DIASTOLE TIME INTERVALS

```
0194    MOV     A,B
        LXI     B,00
        RLC                     · Multiply number of increments by
        RLC                     · 8 and add to value in 3E13, 3E14
        RLC
        LHLD    3E13
        MOV     C,A
        DAD     B
        SHLD    3E13
01A3    LHLD    3E0C            · To compute diastole subtract
        SHLD    3E15            · systole from total count
```

```
        LXI     B, 3E15
        LXI     H, 3E13
        MVI     D,2
        XRA     A
01B2    LDAX    B
        SBB     M
        STAX    B
        DCR     D
        JZ      01BE            · If 2 bytes done go to 01BE
        INX     B
        INX     H
        JMP     01B2
01BE    LDA     3E16
        ADI     03H
        STA     3E16
        MVI     A, 02
        STA     3E10
```

OUTPUT AFTER DELAY

```
01CB    LDA     3E13            · Output LSB and MSB
        OUT     <09>            · Of systole to counter 1
        LDA     3E14            · Output LSB and MSB
        OUT     <09>            · Of diastole to counter 2
        LDA     3E15
        OUT     <0A>
        LDA     3E16
        OUT     <0A>
        LXI     H, 07D0         · Count down from 07D0
01E2    DCX     H               · to zero to obtain 15 ms delay
        JNZ     01E2
        XRA     A               · Trigger systole through
        OUT     <03>            · bit PC3 of 8155
        MVI     A, 08
```

```
        OUT     <03>
01ED    POP     H               • Restore all registers
        POP     D               • and return
        POP     B
        POP     PSW
        RET
```

CONTINUATION OF INITIALIZATION

```
0600    MVI     A,72            • Mode control word counter 1
        OUT     <0B>            • Operates in Mode 1 binary
        MVI     A,B2            • Mode control word counter 2
        OUT     <0B>            • Operates in mode 1 binary
        MVI     A,78            • Load initially systole, diastole
        OUT     <09>            • corresponding to a HR 80 beats/min.
        MVI     A,0A            • into counter 1 and counter 2
        OUT     <09>
        MVI     A,F8
        OUT     <0A>
        MVI     A,0C
        OUT     <0A>
        MVI     A,30            • Mode control word counter 0
        OUT     <0B>            • operates in mode 0, binary
        MVI     A,FF            • initialize count
        OUT     <08>
        OUT     <08>
        XRA     A               • Trigger Systole through
        OUT     <03>            • bit PC3 of 0155
        MVI     A,08
        OUT     <03>
        EI
        HLT
```