A THREE-CHANNEL MICROCOMPUTER SYSTEM FOR QUANTITATIVE ANALYSIS OF THE PHONOCARDIOGRAM, ELECTROCARDIOGRAM, AND CAROTID PULSE SIGNALS

by

Richard J. Lehner

A thesis
presented to the University of Manitoba
in partial fulfillment of the
requirements for the degree of
Master of Science
in
Department of Electrical Engineering

Winnipeg, Manitoba

A THREE-CHANNEL MICROCOMPUTER SYSTEM FOR QUANTITATIVE ANALYSIS OF THE PHONOCARDIOGRAM, ELECTROCARDIOGRAM, AND CAROTID PULSE SIGNALS

BY

RICHARD J. LEHNER

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ABSTRACT

Most diseases of the heart cause changes in the heart sounds and additional murmurs long before other symptoms appear, and hence heart sound analysis by auscultation is the primary test conducted by physicians. The heart sound signal (or phonocardiogram), however, has much more information, which, if analyzed quantitatively in a proper manner could lead to important diagnostic results. A microprocessor-based system is proposed to perform quantitative analysis of the phonocardiogram. The heart sounds are recorded from locations on the chest along with the electrocardiogram (ECG) and carotid pulse signals. The phonocardiogram is segmented into systole and diastole using the ECG and carotid pulse as references. The phonocardiogram is quantified into four parameters representing the time and frequency domain characteristics of the signal. Results of the application of the methods to 47 phonocardiogram signals are presented. This work shows that the parameters of phonocardiogram signals with systolic or diastolic murmurs differ from those of normal signals and hence aid in the detection of murmurs.

ACKNOWLEDGMENTS

I would like to thank Dr. Rangaraj Rangayyan for providing guidance and encouragement and for acting as my thesis supervisor. Dr. Rangayyan's dedication and support is sincerely appreciated.

I would also like to thank Dr. George Collins, Laverne Hastman, R.T. and the staff of the Variety Childrens Heart Centre at the Health Sciences Centre. These people provided an invaluable resource of experience and expertise. The cooperation of the many patients is also acknowledged. Thanks also to Messrs. Jack Sill and Ken Biegen of the Electrical Engineering Department for their assistance in the hardware aspects of the project. I am indebted to many colleagues for their help, particularly Messrs. Jim Slipec, Randall Dembowski, Ken Lam, Dwayne Burek, and J. Winston Brown.

The financial support of this research by the Natural Sciences and Engineering Research Council is also gratefully acknowledged.

CONTENTS

ABSTRACT v

ACKNOWLEDGMENTS vi

Chapter page

I. INTRODUCTION 1

Proposed Method 5
Thesis Outline 7

II. HEART SOUNDS AND PHONOCARDIOGRAPHY 8

Cardiac Physiology 8
The Electrocardiogram 8
The Cardiac Cycle 9
Normal Heart Sounds 12
Murmurs 15
The Carotid Pulse 16
Phonocardiography 20
Auditory Perception of Heart Sounds 20
Relationship Between Heart Sounds, ECG, and Carotid Pulse 21
PCG and Pulse Recording Procedures 22
Review of PCG Signal Processing Techniques . . . 25
Review of ECG and Pulse Signal Processing Techniques 28

III. SYSTEM HARDWARE DESCRIPTION 30

Introduction 30
MC68000 Educational Computer Board 31
Interfacing to the ECB 35
ECB-to-SAAB Interface 37
Data Acquisition System 39
Intel 8231A Arithmetic Processing Unit 48
Digital-to-Analog Conversion Circuitry 53

IV. SIGNAL PROCESSING TECHNIQUES AND SOFTWARE DESCRIPTION 56

Introduction 56
System Initialization 60
Analog-to-Digital Conversion 61
Detection of the QRS-Complex in the ECG 64

Detection of the Dicrotic Notch in the Carotid Pulse 68
PCG Signal Processing 72
Segmentation of PCG into Systole and Diastole 72
PCG Energy Curve 79
PCG Energy Spectrum 80
PCG Energy Distribution Coefficient 87
Summary 88

V. DISCUSSION OF RESULTS 90

VI. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY 106

Conclusions 106
Recommendations 107

REFERENCES 109

Appendix | page

A. SAAB SCHEMATIC DIAGRAMS 119

B. SOFTWARE LISTING 126

LIST OF TABLES

Table		page
3.1.	Educational Computer Board Memory Map	34
3.2.	MC68230 Timer Address Map	34
3.3.	MC68000 Memory Enable Signal Address Map	36
3.4.	Data Acquisition Control Signal Address Map	41
3.5.	Function Table for Multiplexer Control Signals	43
3.7.	APU Function Table	51
3.8.	D/A Control Signal Address Map	54
4.1.	Delay Between Second Heart Sound and Dicrotic Notch	77
4.2.	Typical Execution Times of the PCG Analysis System Routines	89
5.1.	EDC Errors Incurred From Segmentation Errors	94
5.2.	Average EDC_t and EDC_f Values	103
5.3.	Mean and SD of EDC_t and EDC_f Values	105
A.1.	Device Reference	120
A.2.	Device Reference	121

LIST OF FIGURES

Figure		page
2.1.	Typical Normal ECG Lead	11
2.2.	Anterior View of the Opened Heart	11
2.3.	The Etiology of Heart Sounds	14
2.4.	Phonocardiogram of Patient with Severe Aortic Stenosis	17
2.5.	PCG of Patient with Mitral Stenosis	18
2.6.	Typical Normal Carotid Pulse Tracing	18
2.7.	Normal Phonocardiogram, ECG, and Carotid Pulse Tracings	19
2.8.	Locations For Surface Heart Sound Pickups	24
2.9.	Block Diagram of the HP1514B ECG/Phono System . .	24
3.1.	Block Diagram of PCG Analysis System	32
3.2.	MC68000 Memory Enable Signal Decode Logic	36
3.3.	ECB DTACK* Signal Generation	38
3.4.	Block Diagram of Data Acquisition System	40
3.5.	Data Acquisition Control Signal Decode Logic . . .	40
3.6.	Active Lowpass Butterworth Filter	41
3.7.	Schematic Diagram of LF398 Sample-Hold Unit . . .	43
3.8.	Schematic Diagram of CD4051 Analog Multiplexer . .	44
3.9.	Schematic Diagram of ADC1210 A/D Converter	46
3.10.	A/D Operational Timing Diagram	47
3.11.	ADTACK* Signal Generation	47
3.12.	ADTACK* Timing Diagram	48

3.13. Schematic Diagram of INTEL 8231A APU 50

3.14. APUDTACK* Signal Generation 51

3.15. APUDTACK* Timing Diagrams 52

3.16. D/A Circuitry Decode Logic 54

3.17. Schematic Diagram of DAC1210 D/A Converter 55

4.1. Signal Processing Flowchart 59

4.2. MC68000 Status Register 61

4.3. Analog-to-Digital Conversion Flowchart 63

4.4. ECG and ECG Transform g(n) 66

4.5. Carotid Pulse and ECG Transform g(n) 67

4.6. Carotid Pulse and Transform s(n) 71

4.7. PCG signal and Corresponding ECG Transform g(n) 75

4.8. PCG signal and Corresponding Carotid Transform s(n) 76

4.9. PCG Segmented into Systole and Diastole 78

4.10. FFT FORTRAN Source Code 84

4.11. Square Wave and Corresponding 1024-point DFT 85

4.12. Pulse and Corresponding 1024-point DFT 86

5.1. Normal PCG with Simulated Segmentation Errors . . 92

5.2. PCG with Systolic Murmur and Simulated Segmentation Errors 93

5.3. Normal PCG and Corresponding Energy Curves and Energy Spectra 95

5.4. PCG with VSD and Corresponding Energy Curves and Energy Spectra 97

5.5. PCG with PEM and Corresponding Energy Curves and Energy Spectra 98

5.6. PCG with MI and Corresponding Energy Curves and Energy Spectra 99

5.7. PCG with AS and Corresponding Energy Curves and Energy Spectra 100

A.1. SAAB Schematic Diagram (sheet 1 of 3) 122

A.2. SAAB Schematic Diagram (sheet 2 of 3) 123

A.3. SAAB Schematic Diagram (sheet 3 of 3) 124

A.4. USER ROM Schematic Diagram 125

Chapter I

INTRODUCTION

After a half-century of steady increases in heart disease fatalities in the United States, the rate now appears to be on the decline. Unfortunately, heart disease is still by far the nation's leading killer. Cardiologists admit they aren't sure why the death rates are falling. Most of them agree, however, that technology is aiding in the process [1].

The advent of a new generation of early diagnostic technologies such as coronary angiography, digital radiography, computed tomography, radio-nuclide emission imaging, nuclear magnetic resonance (NMR) imaging, and ultrasound techniques [2] are partly responsible for the decline. Presently, coronary angiography is the "gold standard of coronary diagnosis" despite its invasiveness, its use of small but significant radiation doses, and its slight (0.2% or less) risk of provoking a heart attack [1]. However, angiography is not foolproof. The arteries may be obscured by other organs, and image interpretation is sometimes difficult, especially of the smaller vessels. Ultrasound techniques provide views of the cardiac valves and chambers, and with its real-time capabilities, study of the heart valves in motion is possible.

However, due to its persistent problems in resolution and contrast, ultrasound has almost reached its limits as a cardiac tool [1]. NMR imaging provides images of the heart with high clarity and resolution, although interpretation of the images is still quite difficult. NMR imaging is still in its infancy thus it is still very expensive and is something of a wildcard in diagnostic technology [1].

With the arrival of the early diagnostic technology, many basic and fundamental techniques used in clinical practice have been neglected and abandoned in less-than-perfect stages. In particular, the age-old art of heart sound analysis by auscultation suffers from a lack of quantitative analysis techniques. Much needs to be done in order to bring the analysis of the phonocardiogram to a state comparable to that of the electrocardiogram (ECG), the electromyogram (EMG), or the electroencephalogram (EEG).

Frequently, murmurs or alterations in the heart sounds are the only definitive signs of some types of heart disease, appearing long before stress on the cardiovascular system is sufficient to produce other signs and symptoms [3]. Detection and recognition of heart murmurs is a valuable source of information concerning the functioning of heart valves. Auscultation of the heart is presently the primary test performed by cardiologists to assess the condition of the heart. This technique provides beneficial information to the clinician concerning the functional integrity of the heart. However, the technique is plagued by an insufficient understanding of the genesis of heart sounds and any diagnosis based solely on auscultation is questionable and is, in fact, seldom practised. This is due to the wide diversity of opinion concerning the theories that attempt to explain the origin of heart sounds and murmurs [4-20]. Further, from a study of the physical characteristics of heart sounds and human hearing, it is seen that the human ear is poorly suited for cardiac auscultation, thus limiting the capacity of the physician [21]. In addition, auscultation is a subjective test and is prone to interpreter variations and errors [22,23].

This has led to many attempts towards automated analysis of the heart sound signal using the phonocardiograph. The phonocardiogram (PCG), which is a graphic recording of the heart sounds, allows the physician to compare the temporal relationships between the heart sounds and the mechanical and electrical events of the cardiac cycle. While qualitative descriptors used by physicians, such as 'muffled component' of a sound, 'musical murmur', 'rumble', or 'whiff', may be hard to measure or quantify, other features of the PCG can be measured more accurately by quantitative methods. The important features of the PCG are:

1. Frequency content of murmurs and sounds.
2. Maximum intensity of sounds and intensity pattern of murmurs.

3. Timing sequence of murmurs and sounds.
4. Location of the maximum intensity point and the transmission pattern of murmurs and sounds on the precordium.

The need for noninvasive methods has always been present, and phonocardiography could be of great use in achieving this goal. Even though phonocardiography has been available for more than 60 years it has been mainly used by physicians in a qualitative manner. The PCG contains valuable information, which, if analyzed quantitatively in a proper manner, could lead to important diagnostic results. The PCG is currently used in the diagnosis of cardiac malformations and pulmonary hypertension in infants. Noninvasive and passive methods are needed in pediatric cardiology. The small size of the thorax and the presence of fewer intervening structures aids heart sound analysis in infants.

A few automatic diagnostic systems for the PCG have been developed. Presently, the PCG is being used to test the functional and structural integrity of cardiac prosthetic valves. However, there still exists the need for a simple and efficient method to quantify the PCG signal into parameters aiding the detection of murmurs. A system performing such a task would be extremely useful for screening purposes as well as in routine diagnosis. Such a system would also be useful for further heart sound research.

1.1 PROPOSED METHOD

Since only short segments of the PCG are usually examined, and few real time constraints are involved in processing the data, a microcomputer based system could adequately perform the time and frequency domain analysis techniques encountered in phonocardiographic studies. Furthermore, because of its low cost, the system could be dedicated entirely to PCG applications. In this work, a three-channel prototype system, designed to process the PCG, ECG, and carotid pulse signals, is presented. The microcomputer system, hereafter referred to as the PCG analysis system, is used to perform quantitative analysis of the phonocardiogram for the detection of murmurs.

The Motorola 4 MHz MC68000 microprocessor is the central processing unit (CPU) of the PCG analysis system. The system includes 32K bytes of random access memory (RAM), 32K bytes of read only memory (ROM), and input/output (I/O) facilities to provide interfaces to a terminal and host computer. System hardware is also comprised of a three-channel data acquisition system, an Intel 8231A arithmetic processing unit (APU), and a signal display section using three digital-to-analog (D/A) converters. The D/A converters can be connected to an oscilloscope or strip chart recorder. The approximate cost of this prototype system is $1500.00. The system can be easily expanded if additional analog channels or increased processing capabilities are required. The system software is written in MC68000 macro assembly language.

The PCG analysis system is used to implement methods for the time and frequency domain analysis of the systolic and diastolic segments of the PCG signal. A technique for the segmentation of the PCG into systole and diastole using the ECG and carotid pulse as timing references is carried out. Using a QRS-complex detection algorithm based on a smoothed difference of the ECG and a transformation based on a smoothed second difference of the carotid pulse, the PCG can be segmented into systole and diastole. Quantification of the time and frequency domain characteristics of each PCG segment is conducted by first computing the energy curve. An important point involved in the design of a dedicated PCG analysis system is the implementation of a FFT algorithm in assembly language in order to provide the basic core of spectral analysis techniques. Using the FFT routine, the energy spectrum of the PCG is computed. A quantity known as the Energy Distribution Coefficient (EDC) is used to quantify the energy curve and energy spectrum. The PCG signal can then be represented by the systolic and diastolic time and frequency EDC parameters which give an indication of the location of murmurs in time and the frequency content of the signal.

1.2 THESIS OUTLINE

Following a brief description on cardiac physiology and the production of heart sounds and murmurs, the characteristics of the phonocardiogram are outlined and a review of the various signal processing techniques that have been applied to the analysis of the PCG, ECG, and carotid pulse signals is then given. The system hardware and the signal processing techniques developed for this work are then discussed in detail. Following these, a discussion of the results obtained is given along with conclusions and recommendations for further research. Details of hardware used and program listings are given in the appendices.

Chapter II

HEART SOUNDS AND PHONOCARDIOGRAPHY

2.1 CARDIAC PHYSIOLOGY

2.1.1 The Electrocardiogram

The electrocardiogram (ECG) is a manifestation of the electrical activity within the heart. The ECG signal may be recorded by measuring the potential difference between two points on the body. These two points constitute an ECG lead. The triangular lead arrangement known as Einthoven's triangle is normally used, whereby electrodes are placed on the right arm (RA), the left arm (LA), and the left leg (LL). An electrode is also placed on the right leg (RL) and is grounded or used as the reference voltage. The resulting three leads are lead I, LA to RA; lead II, LL to RA; and lead III, LL to LA. Figure 2.1 shows a normal ECG lead along with the conventional terms used in describing the deflections and intervals in the tracing.

2.1.2 The Cardiac Cycle

To understand the physiological aspects of heart sound production, a brief review on the phases of the cardiac cycle is given [24]. Figure 2.2 is an anterior view of the heart showing the cardiac chambers and valves. The human heart has four chambers. The atria are the receiving chambers of the heart. The systemic veins empty into the right atrium and the pulmonary veins empty into the left atrium. The ventricles are the pumping chambers of the heart. The ventricles receive blood from their respective atria and pump it into the major arteries of the pulmonary and systemic circulations. The right ventricle pumps into the pulmonary artery and the left ventricle into the aorta. The atria and ventricles are separated on the right side by the tricuspid valve and on the left side by the mitral valve. Together, the tricuspid and mitral values are referred to as the atrioventricular (AV) valves. The valves situated at the ventricular outflows are the semilunar valves. The pulmonary valve opens into the pulmonary artery and the aortic valve opens into the aorta.

The period of ventricular relaxation, during which the ventricles become filled with blood, is called diastole and the period of ventricular contraction is called systole. The P wave of the ECG, which occurs during late diastole, represents electrical excitation of the atria. In approximately 100 msec atrial contraction commences, causing a slight rise in both atrial and ventricular pressures. At this stage the AV valves are open. Ventricular depolarization then occurs, marked by the onset of the QRS-complex of the ECG, and ventricular contraction begins shortly. As the ventricles contract, the intraventricular pressure rises, closing the AV valves. The semilunar valves are also closed at this time. This phase, termed the the isovolumic contraction phase continues for about 50 msec. The intraventricular pressure increases until this pressure is greater than the pressure in the pulmonary artery and aorta. At this point the semilunar valves open and blood is ejected from the ventricles.

Ventricular systole lasts approximately 300 msec; the ventricles then begin to relax and their pressures drop rapidly, closing the semilunar valves. After semilunar valve closure, the isovolumic relaxation phase continues for about 80 msec and ends with the opening of the AV valves, occurring when ventricular pressures fall below atrial pressures. The phase of rapid filling begins when the AV valves open. During systole, blood accumulates in the atria and the atrial pressures increase. When ventricular pressures become less than atrial pressures, blood moves down from the atria to the ventricles. This process lasts for 100 msec. The final phase of the cardiac cycle is the phase of slow filling or diastasis. This phase lasts about 200 msec and is caused by continued venous return and is terminated by atrial systole.

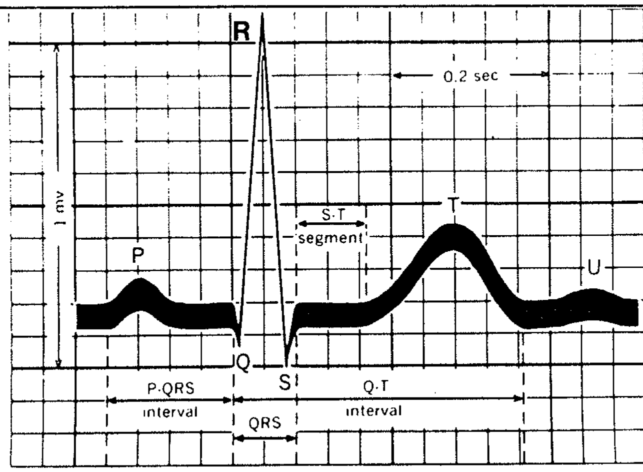


Figure 2.1: Typical Normal ECG Lead

Significant features are the P,Q,R,S, and T waves, the duration of each wave, and time intervals such as the QRS,S-T, and Q-T intervals (From reference [24], p.52)

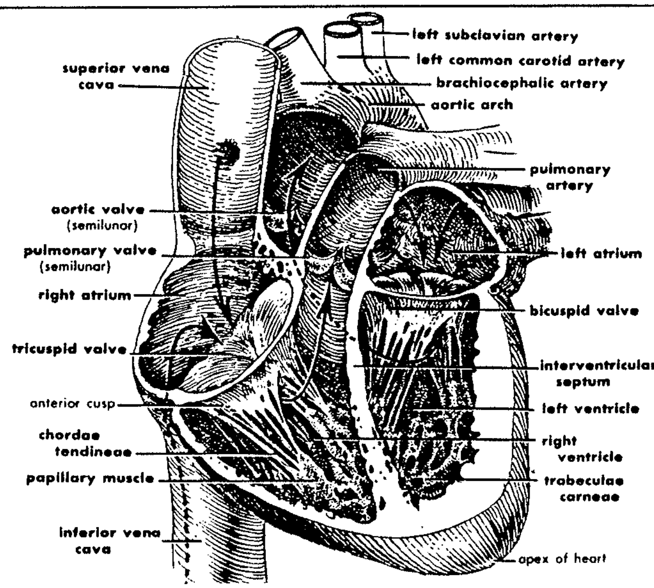


Figure 2.2: Anterior View of the Opened Heart

This figure shows the cardiac chambers and valves. The arrows indicate direction of flow through the heart. (From reference [24], p.28).

2.1.3 Normal Heart Sounds

Heart sounds are generally believed to be caused by the acceleration or deceleration of blood in the heart chambers [3]. There are two major heart sounds that occur during the sequence of one complete cardiac cycle (Figure 2.3).

The first heart sound (S1) can be divided into four components. The first component arises at the onset of ventricular contraction when blood is accelerated in the ventricle, surging blood toward the atrioventricular valves [3]. The frequency and intensity of the first component are very low since the ventricles are relaxed and the acceleration of blood is not high. This movement of blood closes and applies tension to the atrioventricular valves before ventricular pressure rises.

The second component of the first heart sound begins with abrupt tension of the closed AV valves decelerating the moving blood [3]. The frequency of these vibrations is greater than that in the first component. The intensity of the vibrations is dependent upon the velocity of the blood and the abruptness with which it is decelerated.

The third component of the first heart sound occurs during ventricular contraction. The pressure in the ventricle rises above that in the corresponding artery and blood moves toward the semilunar valves [3]. The frequency and intensity of these vibrations is similar to those produced in the second component. The fourth component probably represents vibrations due to turbulence in blood flowing rapidly through the ascending aorta and pulmonary artery [3].

The closure of the semilunar valves gives rise to the second heart sound (S2). Although the primary vibrations occur in the arteries, they are also transmitted to the ventricles and atria by movement of the blood, valves, and valve rings [3]. The frequency of the second sound is usually higher than that of the first sound and is normally split into two components (A2,P2), reflecting the fact that the aortic valve normally closes before the pulmonary valve. Pathological conditions may cause this gap to widen, or may also reverse the order of occurrence of A2 and P2.

In some cases a third heart sound (S3) may be heard, corresponding to sudden termination of the ventricular rapid filling phase. These vibrations are of low frequency and the sound is generally inaudible in the normal adult, but is frequently heard in children. In late diastole, a fourth heart sound (S4) may be sometimes heard, caused by atrial contractions displacing blood into the distended ventricles. In addition to these sounds, valvular clicks and snaps are occasionally heard.

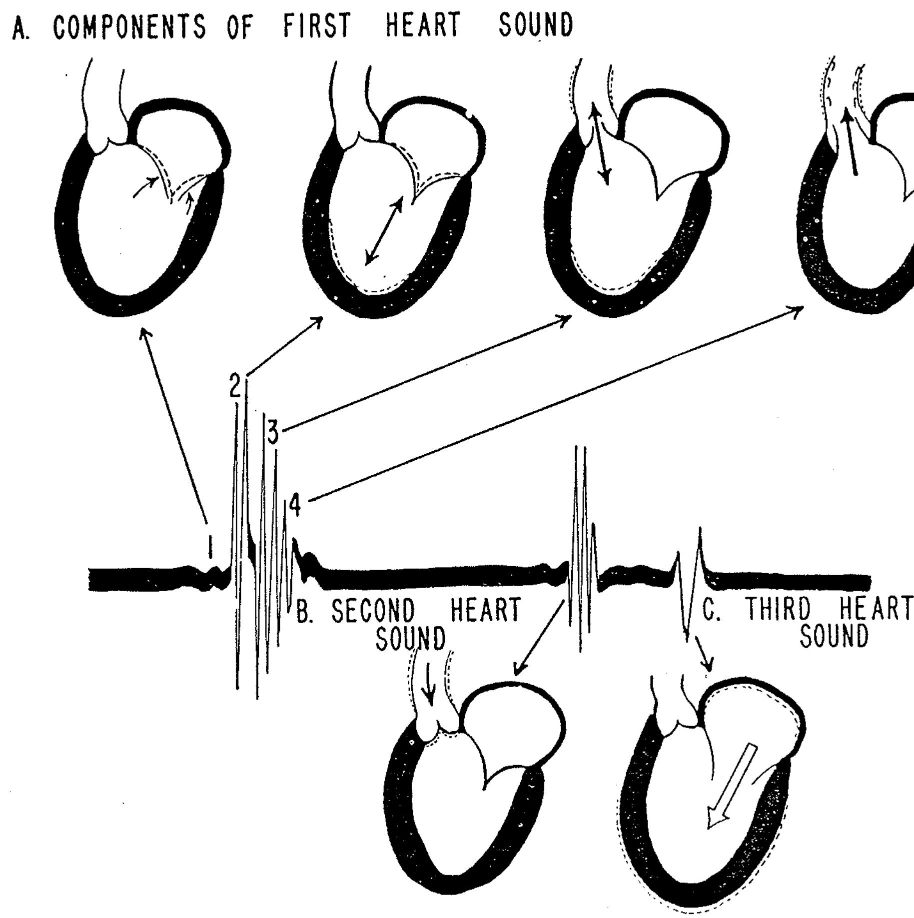


Figure 2.3: The Etiology of Heart Sounds

Schematic drawing of the causes of various components of the heart sounds based on the concept that the vibrations are induced by acceleration or deceleration of the blood within elastic chambers. (From reference [3], p.424).

A. The first heart sound can be divided into 4 components. The initial vibrations occur when the first myocardial contractions in the ventricle shift blood toward the atrium to seal the AV valves. The second component begins with abrupt tension of closed AV valves decelerating the blood. The third component involves oscillations of blood between the distending root of the aorta and the ventricular walls. The fourth component represents vibrations due to turbulence in blood flowing rapidly through the ascending aorta and pulmonary artery.

B. The second heart sound begins with closure and tensing of the semilunar valves. Although primary vibrations occur in the arteries, they are also transmitted to ventricles and atria by movement of blood, valves, and valve rings.

C. The third heart sound occurs at the end of the rapid filling phase. Sudden termination of this phase may throw the entire AV system into vibrations which have very low frequency.

2.1.4 Murmurs

The intervals between S1 and S2, and S2 and S1 of the next cardiac cycle (corresponding to ventricular systole and diastole) are normally silent. Murmurs, which are caused by certain cardiovascular defects and diseases, occur in these intervals. Murmurs are high frequency noise-like sounds which arise when the blood velocity becomes high in the presence of an irregularity through which the blood flows. Typical conditions in the cardiovascular system which cause blood flow turbulence include valvular stenosis and valvular insufficiency.

Systolic murmurs (SM) are caused by conditions such as ventricular septal defect (VSD), aortic stenosis (AS), pulmonary stenosis (PS), mitral insufficiency (MI), and tricuspid insufficiency (TI). While semilunar stenosis (AS,PS) causes an obstruction in the path of blood being ejected during systole, AV valvular insufficiency (MI,TI) causes regurgitation of blood to the atria during ventricular contraction. Diastolic murmurs (DM) are caused by such conditions as aortic and pulmonary insufficiency (AI,PI) and mitral and tricuspid stenosis (MS,TS). Other conditions causing murmurs are atrial septal defect (ASD), patent ductus arteriosus (PDA), as well as certain functional conditions due to vigorous exercises, which increases cardiac output and blood velocity.

Although murmurs are noise-like, certain features aid in distinguishing between different causes. For example, AS causes a diamond shaped midsystolic murmur, while MS causes a decrescendo-crescendo type diastolic-presystolic murmur. See Figures 2.4 and 2.5 for tracings of such signals.

2.1.5 The Carotid Pulse

The carotid artery in the neck is the point of the arterial system closest to the heart where external sensing can be performed. The carotid pulse tracing (Figure 2.6) begins to rise abruptly with aortic ejection and reaches its initial peak at the time ejection is probably at its maximum [25]. The first peak, called the percussion wave (P), is usually followed by a plateau or secondary wave, called the tidal wave (T), late in systole [25]. The pulse then falls smoothly to a point in which the dicrotic notch (D) is inscribed. The tidal wave represents primarily the reflected pulse returning from the upper body. The dicrotic notch is produced by abrupt completion of aortic valve closure [25]. In early diastole a small positive wave designated as the dicrotic wave (DW) occurs, which represents the reflected pulse from the lower body [25]. Figure 2.7 shows the important intervals of the carotid waveform.

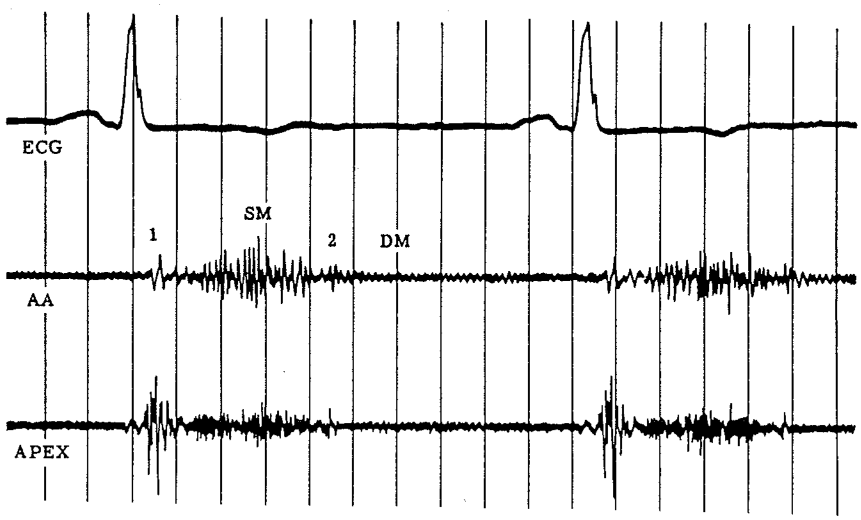


Figure 2.4: Phonocardiogram of Patient with Severe Aortic Stenosis

The ECG signal is also shown. Note the systolic murmur with peak intensity in midsystole (diamond shaped envelope). (From reference [25], p.132).

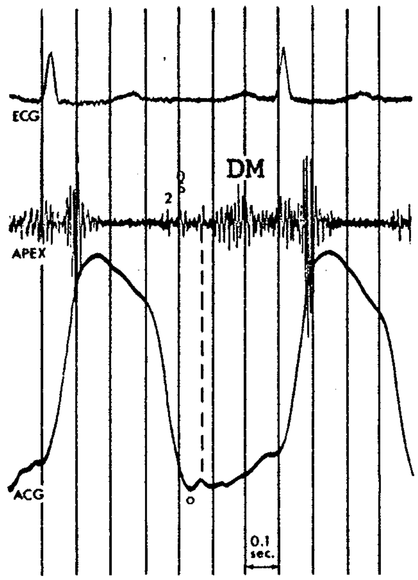


Figure 2.5: PCG of Patient with Mitral Stenosis

The ECG signal and apexcardiogram are also shown. Note the pre-systolic accentuation of the diastolic murmur. (From reference [25], p.152).

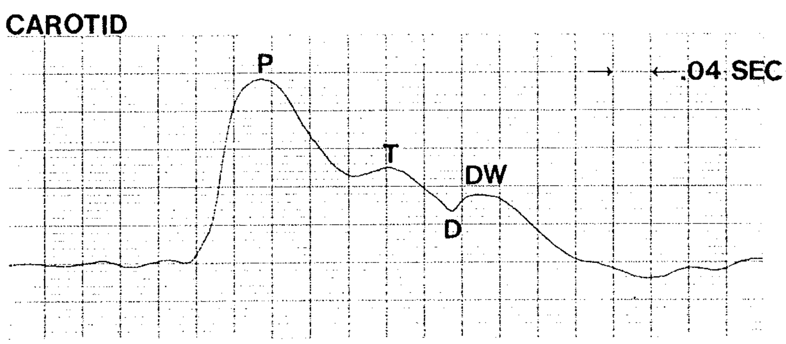


Figure 2.6: Typical Normal Carotid Pulse Tracing

Percussion wave P occurs approximately at maximum carotid ejection. Tidal wave T follows late in systole. Dicrotic notch D indicates closure of aortic valve. In diastole the dicrotic wave DW occurs.

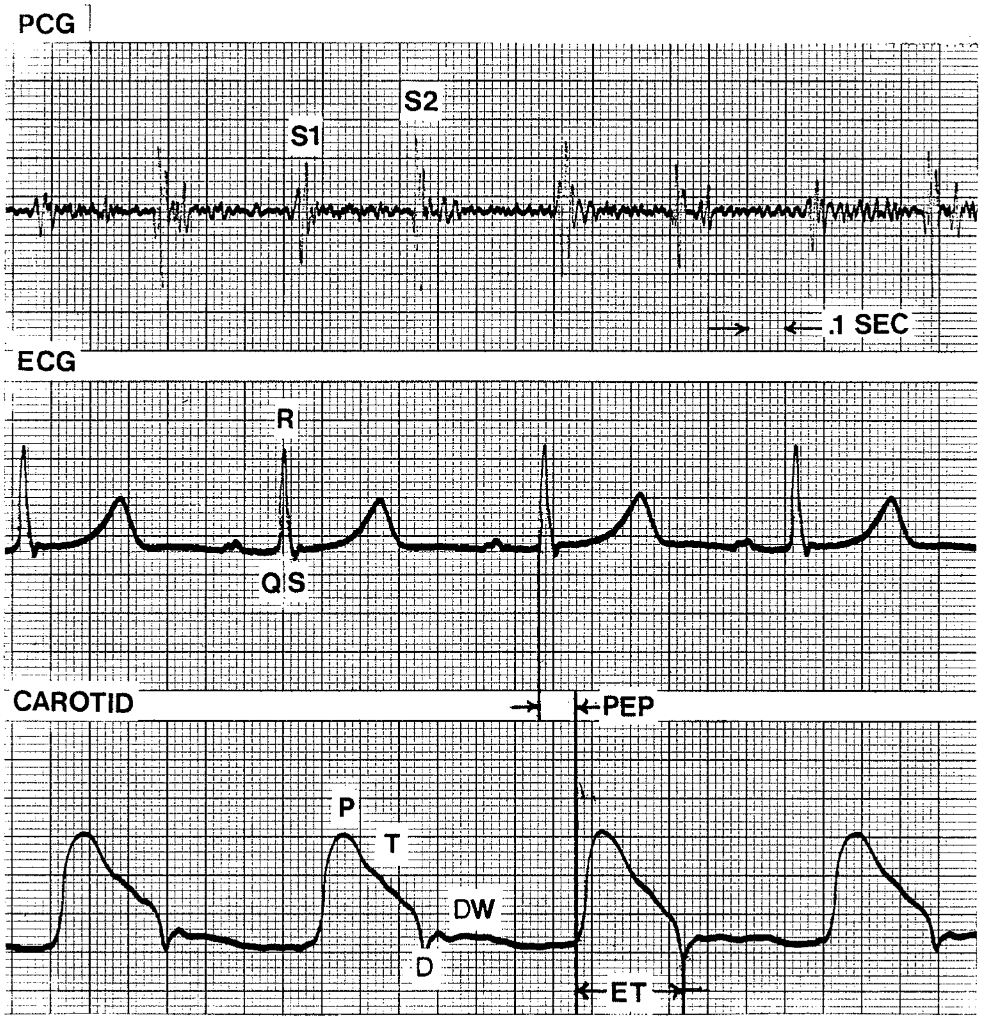


Figure 2.7: Normal Phonocardiogram, ECG, and Carotid Pulse Tracings

The pre-ejection period PEP is measured from the Q wave of the ECG to the onset of carotid upstroke. The ejection time ET is the interval from the start of carotid upstroke to the dicrotic notch. Note that the first heart sound occurs at the end of the QRS complex. The second heart sound occurs just before dicrotic notch.

2.2 PHONOCARDIOGRAPHY

2.2.1 Auditory Perception of Heart Sounds

Due to the varying threshold of audibility of the human auditory system with frequency of sound presented, and the low frequency nature of heart sounds, only a portion of the cardiohemic vibrations is audible [3]. The maximum sensitivity of human auditory perception lies in the 1000 Hz to 2000 Hz region, which is above the usual range of cardiac sounds. Normal heart sounds usually lie in the 20 Hz to 200 Hz range while the maximum frequency of murmurs is 600 Hz [3].

From a study of the physical characteristics of heart sounds and human hearing, it is seen that the human ear is poorly suited for cardiac auscultation [21]. Certain details such as the presence of murmurs, which are of much higher frequency than the normal sounds, can be detected easily by auscultation. However, it has been shown that auscultation is a subjective test and is prone to interpreter variations and errors [22,23].

Although the training and experience of the cardiologist do improve results attained by auscultation, the heart signal has much more information. This has led to many attempts towards automated analysis of the heart sound signal using the phonocardiograph.

2.2.2 Relationship Between Heart Sounds, ECG, and Carotid Pulse

The phonocardiograph [26], which consists of microphones, selective filters and amplifiers, and a recording unit, gives a graphic recording of heart sounds and murmurs (called a phonocardiogram or PCG). It is valuable in that it eliminates the subjective interpretation of these sounds. Additional channels are also provided for the recording of reference tracings so that the evaluation of heart sounds and murmurs with respect to the electrical and mechanical events in the cardiac cycle can be conducted.

The relationship between the PCG, ECG, and carotid pulse can be seen in Figure 2.7. The ECG can be used as a reference in identifying the first heart sound. The beginning of the QRS-complex provides a sharp reference which can be used to accurately determine the onset of ventricular contraction. Thus the R wave can be used to determine the start of the first heart sound.

Although variable, the beginning of the steep ascent of the carotid pulse usually coincides with the fourth component of the first heart sound [25]. The dicrotic notch follows A2 (aortic component of the second heart sound) by the time required to travel to the recording site in the neck. This usually amounts to 10 - 50 msec and is dependent upon the distance of the recording site from the heart and the pulse-wave velocity [25].

2.2.3 PCG and Pulse Recording Procedures

Since heart sounds and murmurs are of low amplitude, extraneous sounds must be minimized in the vicinity of the patient. Thus it is a standard procedure to record the phonocardiogram in a room that is as quiet as possible. The patient is placed in a supine position with the head on a pillow. ECG leads are placed on all four extremities and a standard Lead II is recorded throughout the procedure. There are optimal recording sites for various heart sounds, sites at which the intensity of sound is the highest because the sound is being transmitted through solid tissue or through a minimal thickness of inflated lung [27]. An HP21050A contact sensor is strapped firmly to the patient at one of the heart sound pickup areas on the chest shown in Figure 2.8. These locations are as follows:

1. Aortic area: second right intercostal space.
2. Pulmonary area: second left intercostal space.
3. Tricuspid area: fourth left intercostal space.
4. Mitral area: near the apex of the heart.

The HP21050A contact sensor has a relatively flat response from 0.05 Hz to 1000 Hz with an equal attenuation of about 5 dB over the entire range. This makes it equally suitable to record the first and second heart sounds (below 200 Hz) and high frequency murmurs (up to 600 Hz).

The HP21281A pulse transducer kit is used to record the carotid pulse. A trained technician locates the site where the carotid pulsation in the neck is the strongest and the pickup pressure cup is applied firmly. This transducer has a frequency response of approximately dc to 1000 Hz.

A three-channel HP1514B ECG/Phono system is used to obtain an amplified and filtered phonocardiogram with simultaneous ECG and carotid pulse reference signals. Channel 1 records a filtered heart sound. In this work, the PCG was obtained using a highpass filter (cutoff frequency of 25 Hz). Channel 2 is used to record a Lead II ECG as a timing reference. Any lead could be used as long as a suitable reference QRS-complex is obtained. Channel 3 is used to record a highpass filtered carotid pulse with a cutoff frequency of 1 Hz. Figure 2.9 shows a block diagram of the HP1514B ECG/Phono recording system as it is used to record the PCG and carotid pulse in this project.

A four-channel HP3960 FM instrumentation recorder is used to simultaneously record the PCG, ECG, and carotid pulse waveforms. The signals are recorded for a duration of 10 seconds.[1]

---

[1] The recording time of 10s is sufficient to diagnose heart diseases from the PCG. It is desirable to reduce recording time to a few seconds, especially for children, since the patient's breath is held during recording.

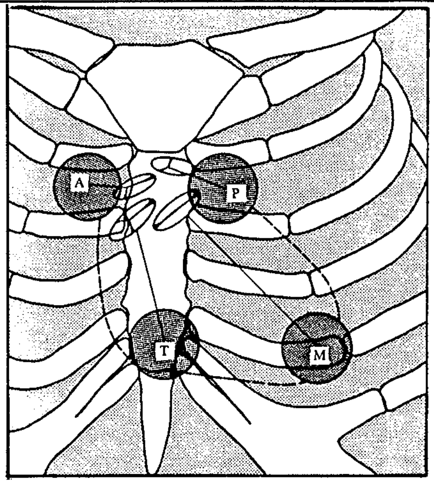


Figure 2.8: Locations For Surface Heart Sound Pickups

A, aortic area: second right intercostal space; P,pulmonary area: second left intercostal space; T, tricuspid area: fourth left intercostal space; M, mitral area: near apex of heart. (From reference [28], p.357)

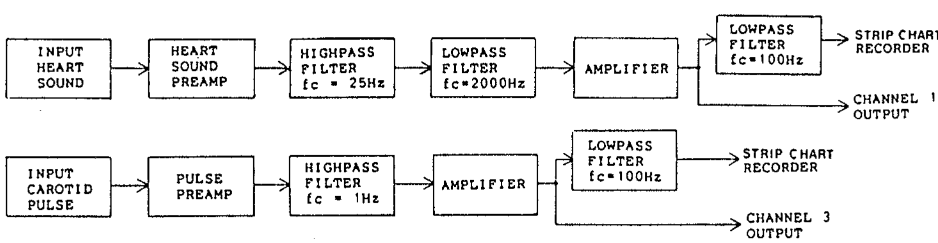


Figure 2.9: Block Diagram of the HP1514B ECG/Phono System

Note that the heart sound signal and carotid pulse are highpass filtered at 25 Hz and 1 Hz, respectively. The signals are lowpass filtered at 100 Hz and directed to a strip chart recorder. The unfiltered signals are directed to the respective output channels.

2.3 REVIEW OF PCG SIGNAL PROCESSING TECHNIQUES

Many signal processing techniques have been tried to quantify the PCG signal. The time-envelope of energy of the PCG signal is clinically significant as murmurs of specific envelope shapes occur in certain parts of the cardiac cycle. For example, aortic stenosis causes a diamond shaped systolic murmur, while mitral stenosis leads to a decrescendo-crescendo type diastolic-presystolic murmur. Early techniques to extract the sound envelope employed demodulation and synchronized averaging of the envelope of the cardiac sounds [29]. Complex demodulation and power spectrum estimation techniques have been conducted to extract the 'envelogram' of the PCG signal [30]. In another method, the average power in contiguous 10 msec segments was plotted to obtain a 'power curve' [31]. Maximum amplitude and zero crossing measurements of the PCG signal have been attempted [32]. The use of wave analyzers and bandpass filter banks to improve detection of aortic stenosis in the presence of mitral insufficiency has also been reported [33]. In addition, cardiac screening systems have been designed based on measurements of frequency content, amplitude, energy, and timing of heart sounds [34-36].

Analysis of the PCG signal in the frequency domain has also been extensively researched [37-52]. A sound spectrograph, in which sound intensity is displayed in gray shades against time and frequency coordinates is discussed in reference [37]. Three dimensional contour plots of the PCG involving contours of equal intensity against time and frequency coordinates have been conducted [38]. Fast Fourier transform analysis of PCG signals has been attempted [42-49]. Other techniques of spectral analysis on the PCG are discussed in references [50-52].

Linear predictive analysis has been used to extract the spectral patterns from the PCG signal [53]. These parameters are used as features for pattern classification. The linear predictive method was also used to derive an algorithm for detection and segmentation of the first and second heart sounds based on the frequency domain characteristics of the heart sounds [54]. A spectral analysis technique using selective linear prediction coding to determine the spectral distribution of the second heart sound is described in reference [55].

In addition, images of sound distribution produced from multiple PCGs recorded on the chest wall have been used to compute chest wall maps showing the varying distribution patterns of sounds and murmurs over the passage of time [56]. Statistical analysis of heart sounds has been performed using the first two statistical moments to classify the heart sounds into deterministic and nondeterministic sounds and to segment the heart sound components [57,58].

The phonocardiogram has also been used to test the functional and structural integrity of cardiac prosthetic valves [59-68]. Parametric pole-zero modelling methods for frequency domain feature extraction [69] and FFT techniques for spectral analysis [70,71] have been reported for prosthetic valve sound analysis.

Digital signal processing techniques such as homomorphic filtering have been applied to the PCG with limited success [72-75]. Pole-zero modelling by Shanks' method was also attempted with some success [72,74-76]. Procedures for quantitative analysis of the PCG in the time and frequency domain have been proposed using the energy curve and power spectrum [77,78,79]. The application of speech signal processing techniques to the analysis of the PCG has been discussed in reference [80].

An overview of the methods tried illustrates that there have been many attempts at the analysis of the PCG. However, there still exists a need for a simple and efficient method to quantify the subjective features of the PCG and classify the various types of murmurs. A system performing such a task, realizable for online analysis, would be extremely useful for screening purposes, as well as in routine diagnosis and further heart sound research.

2.4 REVIEW OF ECG AND PULSE SIGNAL PROCESSING TECHNIQUES

The ECG and carotid pulse can be used as reference signals for the identification of features in the PCG. In particular, the QRS-complex of the ECG provides a sharp reference which can be used to identify the first heart sound (start of systole). The dicrotic notch in the carotid pulse can be used to approximate the onset of the second heart sound (start of diastole).

The application of signal processing techniques to the analysis of the ECG has been quite extensive; see, for example, references [81-86]. A thorough review of the digital analysis of the ECG can be found in reference [87]. Reported techniques have used a transform based on the first difference of the ECG signal as a reference for the PCG [88,89]. A single peak corresponding to the QRS-complex is found and used to identify the first heart sound.

The waveshape for the carotid pulse is essentially the same as for the aortic pulse and, as such, many of the arterial pulse wave analysis techniques reported can be applied directly to the carotid pulse. Reviews of some of these methods are given in references [87] and [90].

The use of specially designed filters that selectively exaggerate the feature sought, as well as minimize noise, has been reported [91]. An application of the AZTEC preprocessing program [92] has been used to initiate the search for important points, such as systolic upstroke and onset of dicrotic notch, in the central arterial pressure wave [93].

Pattern recognition techniques have also been used to identify the features of arterial pulse waves. An algorithm for the automatic identification of the important points in brachial pulse waves is described in references [94,95]. A waveform parsing technique has been used to detect and measure structural variations of the carotid pulse wave [96].

A real time pressure algorithm has been developed using data reduction analysis and specific selectable thresholds to filter noise and detect critical points [97]. A method for the processing of the arterial pressure wave involving a series of algorithms to identify the major events of the aortic pressure wave is reported in [98].

Most reported pulse analysis techniques have been for real-time analysis and are concerned with data reduction techniques and the identification of all major events in the pressure cycle. In this study, real-time processing of the carotid pulse is not necessary. One cardiac cycle of the carotid pulse, in which the dicrotic notch is localized, need only be extracted. A transform based on the second derivative of the carotid pulse is computed to provide a reference point for the identification of the second heart sound.

Chapter III

SYSTEM HARDWARE DESCRIPTION

3.1 INTRODUCTION

The Motorola MC68000 microprocessor is the central processing unit of the prototype PCG analysis system. A detailed operational description of the MC68000 processor is available in the MC68000 User's Manual [99]. The analysis system consists of two computer boards; the Motorola Educational Computer Board (ECB) and a Signal Acquisition and Analysis Board (SAAB). The ECB consists of the microprocessor, memory, timer, and input/output (I/O) facilities. The user interface to the ECB is provided by connecting the board to a terminal via an RS-232 interface. A second serial port is provided to connect the ECB to a host computer system. A complete hardware description of the ECB is provided in the ECB User's Manual [100].

The SAAB, designed to be directly interfaced to the ECB, consists of the following:

1. A three-channel data acquisition system including anti-aliasing filters, sample/hold units, an analog multiplexer, and an analog-to-digital (A/D) converter.

2. An arithmetic processing unit (APU) to provide extended signal processing capabilities.
3. Three digital-to-analog (D/A) converters to display the various output signals on an oscilloscope or strip chart recorder.

Figure 3.1 shows a block diagram of the PCG analysis system. The SAAB schematic diagrams are included in Appendix A.

This chapter provides a brief description of the MC68000 Educational Computer Board hardware including the MC68230 Peripheral Interface/Timer and ECB interfacing techniques. Following this a detailed description of the signal acquisition and analysis board will be given including the data acquisition system, the APU and the D/A circuitry. In this discussion, active low signals are designated by an asterisk (*) following the signal name.

3.2 MC68000 EDUCATIONAL COMPUTER BOARD

The ECB includes a 4 MHz MC68000 16-bit microprocessor, 32K bytes of dynamic random access memory (RAM), 16K bytes of firmware read-only memory (ROM), two serial communication ports (RS-232 compatible), a parallel port, a 24-bit programmable timer, and a wire-wrap area.

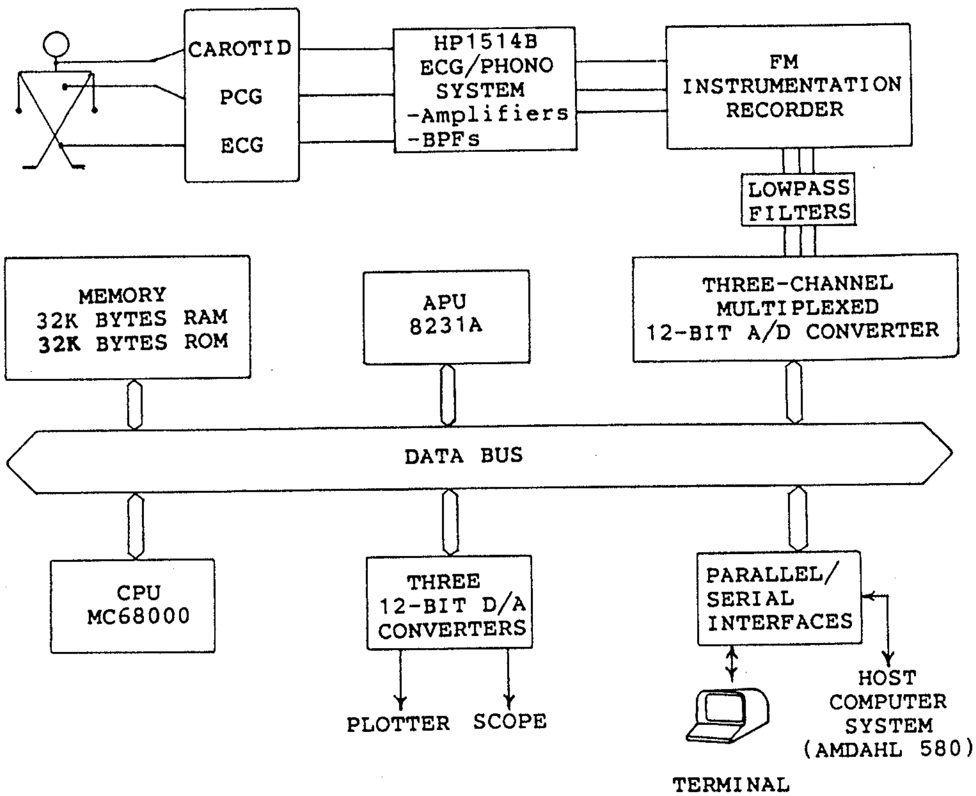


Figure 3.1: Block Diagram of PCG Analysis System

The ECB memory map is shown in Table 3.1. The RAM is addressed at the bottom of the memory map ($000007-$007FFF) and consists of sixteen MCM4116B devices. The user area of RAM extends from $000900-$007FFF. The firmware ROM is located at $008000-$00BFFF. The system firmware is stored in two MCM68A364 devices. All I/O devices are mapped into the same 64K byte page at $010000-$01FFFF. Redundant mapping occurs since the address is not fully decoded (i.e. the same device appears at several addresses).

Two MC6850 asynchronous communications interface adapters (ACIAs) provide the bus interface for the two serial ports. One of the ACIAs is used for connecting a terminal to ECB Port 1 and the other ACIA is used to connect a host computer system to ECB Port 2. A baud rate generator provides the transmit and receive clocks for both ACIAs. Both serial ports are RS-232 compatible.

The ECB includes an MC68230 peripheral interface/timer (PI/T) device. This device consists of two parallel interface ports, 8 general purpose I/O pins, and a 24-bit programmable timer. Table 3.2 shows the address map for the MC68230 timer. For complete details regarding the timer registers and programmable options available to the timer, refer to the MC68230 PI/T Data Sheet [101].

TABLE 3.1

Educational Computer Board Memory Map

Function	Address
System Memory	\$000000-\$0008FF
Vector Table	\$000000-\$0003FF
User Memory	\$000900-\$007FFF
Tutor Firmware	\$008000-\$00BFFF
I/O Devices	\$010000-\$01FFFF
PI/T	\$010000-\$01003F
ACIAs	\$010040-\$010043
Not Used	\$020000-\$02FFFF
M6800 Page	\$030000-\$03FFFF
Not Used	\$040000-\$FFFFFF

TABLE 3.2

MC68230 Timer Address Map

Address	PI/T Register
\$010021	Timer Control Register (TCR)
\$010023	Timer Interrupt Vector Register (TIVR)
\$010027	Counter Preload Register High (CPRH)
\$010029	Counter Preload Register Middle (CPRM)
\$01002B	Counter Preload Register Low (CPRL)
\$01002F	Counter Register High (CNTRH)
\$010031	Counter Register Middle (CNTRM)
\$010033	Counter Register Low (CNTRL)
\$010035	Timer Status Register (TSR)

3.2.1 Interfacing to the ECB

A small wirewrap area is provided directly on the ECB. In addition to the wire wrap area, the ECB has a connection area which gives access to 46 of the MC68000 bus and system timing signals and has provision for an auxiliary 50 pin I/O header.

Five connections on the ECB, denoted E1*-E5*, are provided, giving active-low enable signals for unused areas of the MC68000 memory map. Figure 3.2 shows the decode logic for these enable signals and Table 3.3 shows the memory map for the signals.

When interfacing to the ECB, special care must be taken with the Data Transfer Acknowledge signal (DTACK*). This input indicates that the data transfer between the processor and a device is completed. It may be used as a handshake line when the MC68000 is used in asynchronous operations. The processor DTACK* is generated by ANDing DTACK PIT*, DTACK RAM*, and DTACK ROM*, as shown in Figure 3.3. The processor DTACK* goes low whenever any of these go low. Another signal can not be added to the processor DTACK* since the signal is not an open collector output. The USER DTACK* is connected to the system via ECB connection E7. The DTACK PIT* is turned off when not used and the USER DTACK* can be bussed at this point. The USER DTACK* must be an open-collector or three-state driver.

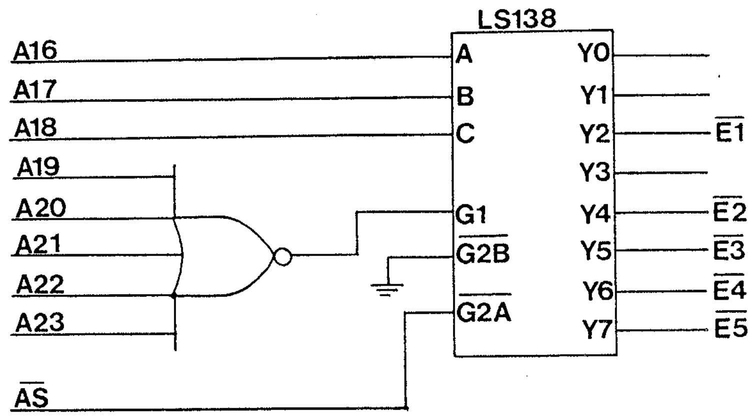


Figure 3.2: MC68000 Memory Enable Signal Decode Logic

TABLE 3.3

MC68000 Memory Enable Signal Address Map

Enable Signal	Address Segment
E1*	\$020000-\$02FFFF
E2*	\$040000-\$04FFFF
E3*	\$050000-\$05FFFF
E4*	\$060000-\$06FFFF
E5*	\$070000-\$07FFFF

3.2.2 ECB-to-SAAB Interface

Individual wirewrap pins are soldered into the connections provided on the ECB. Signals from these pins are wire-wrapped to a 50-pin right angle header, which is inserted into the auxiliary I/O header location on the ECB. The SAAB has a similar 50-pin header. The interface between the ECB and the SAAB is provided via a 50-conductor ribbon cable. Four extra signals are connected to the header, these being the memory enable signals E1*, E2*, E3*, and the USER DTACK* connection E7.

The MC68000 data bus and address bus are buffered using 74LS245 and 74LS241 devices, respectively. These buffers are enabled only when devices on the SAAB are addressed and when the MC68000 ADDRESS STROBE (AS*) is asserted, indicating there is a valid address on the address bus. The MC68000 READ/WRITE (R/W*) signal is used to define the data bus transfer as a read or write cycle.

The SAAB USER DTACK* is provided by ANDing the respective DTACKs* from the data acquisition system, arithmetic processing unit, and digital-to-analog conversion circuitry. This signal goes low indicating completion of data transfer.

The MC68000 microprocessor can handle seven levels of interrupts. Interrupt requests to the ECB are restricted to an M6800 type autovectored priority level 4 interrupt. The ECB provides an autovectored interrupt request designated

6800 IRQ*. The 6800 IRQ* signal is generated by ANDing the interrupt request signals from the data acquisition system (ADIRQ*) and the APU (APUIRQ*). When either of these signals goes low the 6800 IRQ* line is asserted, generating the level 4 interrupt on the MC68000.

The ECB wirewrap area is used to provide 16K bytes of user ROM, using two 2764 devices. The ROM is mapped into memory segment $070000-$07FFFF. Connection E5* from the ECB is used in the decoding scheme. The USEROM DTACK* signal is generated using a 74LS175 device. Schematic diagrams of the ECB-to-SAAB interface and the ROM circuitry are found in Appendix A.

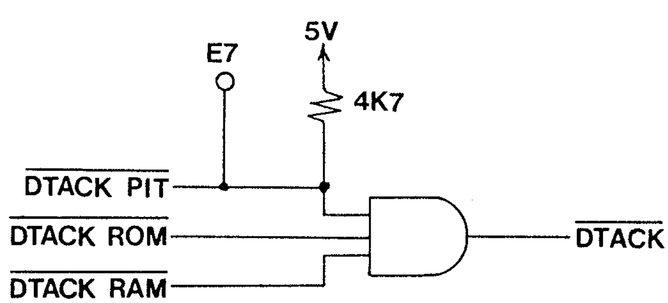


Figure 3.3: ECB DTACK* Signal Generation

3.3 DATA ACQUISITION SYSTEM

A block diagram of the data acquisition system is shown in Figure 3.4. In this system, data is taken from the analog inputs at the same instant of time, requiring a sample-hold unit per channel ahead of the analog multiplexer. All sample-holds are given the hold command simultaneously. The multiplexer then sequentially switches to each sample-hold output while the analog-to-digital (A/D) converter converts the signal into digital form.

The data acquisition system is mapped into memory segment \$020000-\$02FFFF. Data acquisition control is accomplished using a 74LS138 3- to 8-line decoder. Connection E1* from the ECB and MC68000 address lines A1-A3 are used in the decoding scheme, as shown in Figure 3.5. Table 3.4 shows the address map for the data acquisition control signals.

The PCG, ECG, and carotid pulse signals are directed through anti-aliasing filters using the active lowpass Butterworth filter shown in Figure 3.6. These lowpass filters (one per channel) are constructed using MC741 operational amplifiers. The lowpass filter cutoff frequency (f_c) for the PCG signal is 500 Hz. The cutoff frequency for the ECG signal and the carotid pulse is 100 Hz.

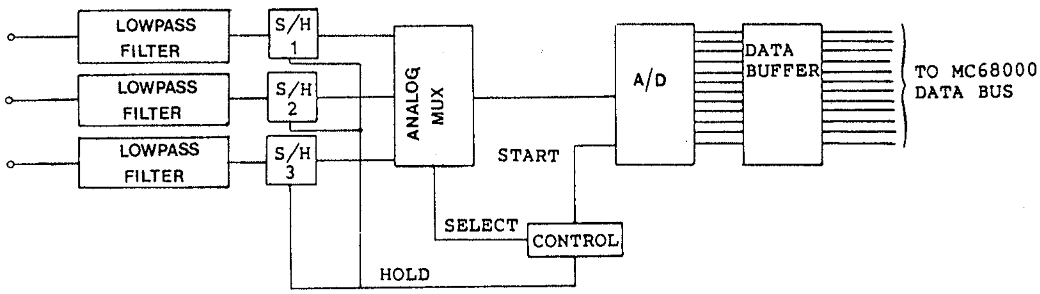


Figure 3.4: Block Diagram of Data Acquisition System

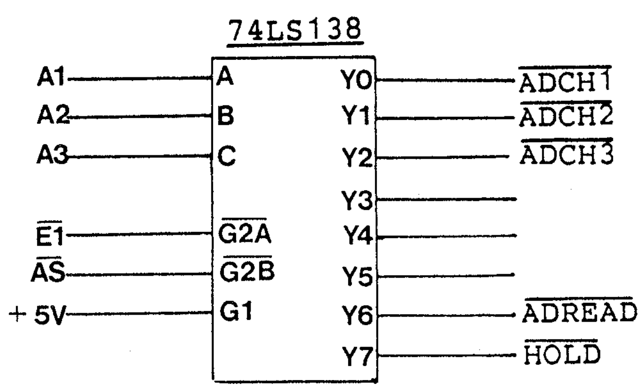


Figure 3.5: Data Acquisition Control Signal Decode Logic

TABLE 3.4

Data Acquisition Control Signal Address Map

Address	Function
$020001	Select ECG channel of multiplexer (ADCH1)
$020003	Select PCG channel of multiplexer (ADCH2)
$020005	Select Carotid channel of multiplexer (ADCH3)
$020006	Not Used
$020008	Not Used
$02000A	Not Used
$02000C	Read A/D Converter (ADREAD)
$02000E	Initiate hold state of S/H devices (HOLD)

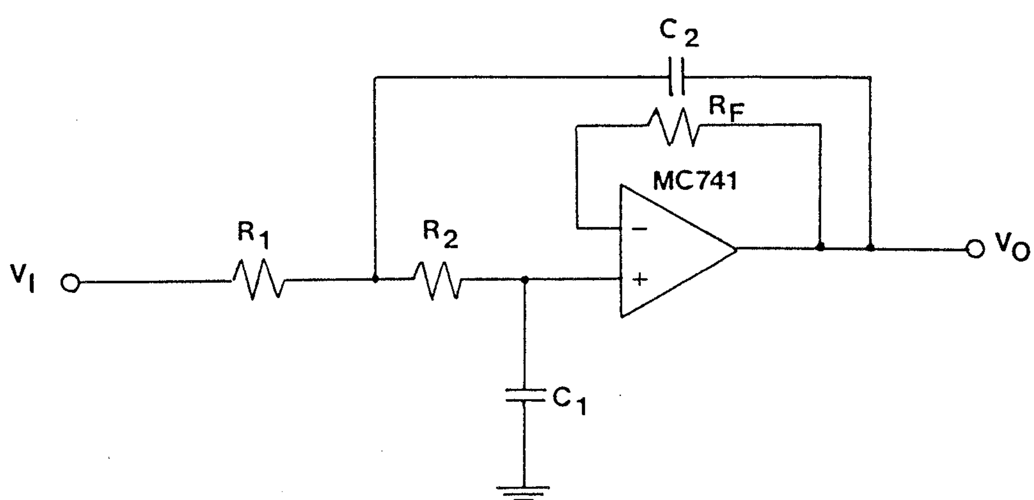


Figure 3.6: Active Lowpass Butterworth Filter

Lowpass filter designed such that $R = R_1 = R_2, R_f = 2R$, $C_2 = 2C_1$ and $C_1 = .707/\omega_c R$.

Three National Semiconductor LF398 Sample and Hold Circuits are used to hold the three signals at the same instant of time. Figure 3.7 shows a schematic diagram of a sample/hold unit as it is configured in the data acquisition system. The hold mode is initiated by taking the logic input of each device high. This is accomplished by asserting the HOLD* signal. The MC555 timer is set up in monostable mode resulting in a 5.0 volt, 30.0 μsec pulse whenever HOLD* is asserted. With a hold capacitor of 0.01 μF, typical acquisition time is 20.0 μsec.

A National Semiconductor CD4051 device is employed in this system as a 3-channel multiplexer with three control inputs A, B, and C. The three control signals select 1 of 3 channels to be turned "ON" and connect the input to the output. The function table for the three select inputs is shown in Table 3.5. A schematic diagram of the multiplexer as it is configured in this system is shown in Figure 3.8. To simplify multiplexer control the filtered ECG, PCG and Carotid pulse signals are connected to the multiplexer input channels 6, 5, and 3, respectively. Control of channel selection is accomplished using a 74LS175 device in conjunction with the ADCH1*, ADCH2*, and ADCH3* signals and the CONVERSION COMPLETE (CC*) signal from the A/D converter.

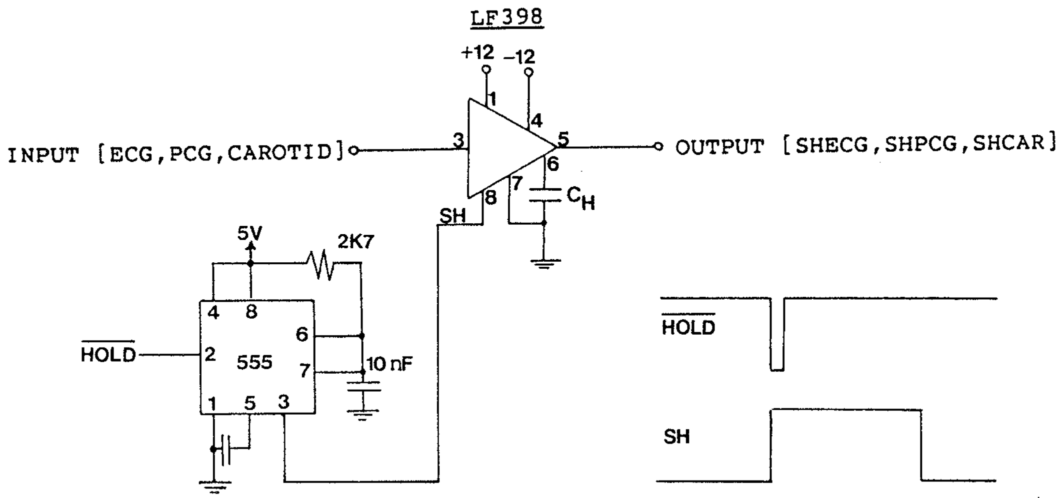


Figure 3.7: Schematic Diagram of LF398 Sample-Hold Unit

TABLE 3.5

Function Table for Multiplexer Control Signals

Input States C	B	A	"ON" Channels
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

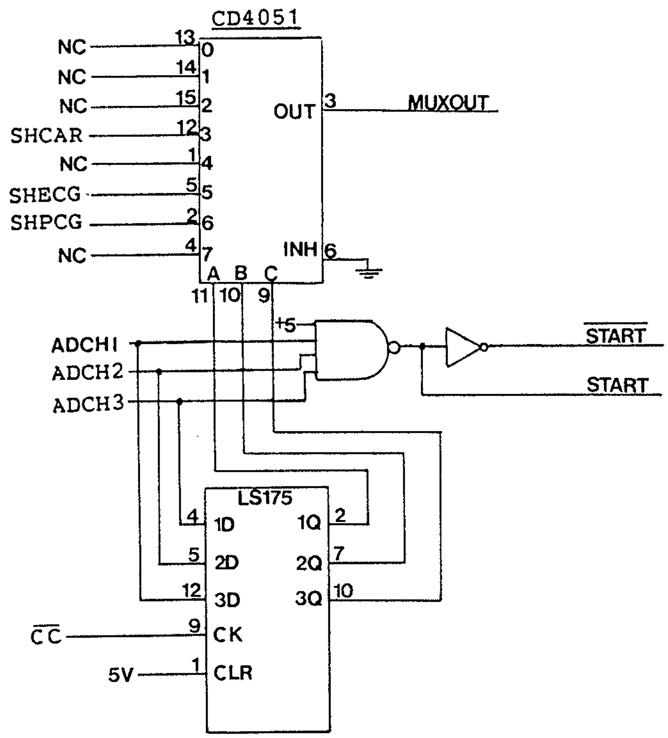


Figure 3.8: Schematic Diagram of CD4051 Analog Multiplexer

The National Semiconductor ADC1210 is a CMOS device used to perform 12-bit successive approximation (SA) analog-to-digital conversion. The A/D converter operates at a clock frequency of 125 KHz. This clock is derived by dividing the ECB 1MHZ CLOCK signal by 8 using a 74LS93 4-bit binary counter. The A/D is configured to convert analog inputs in the $\pm$ 2.5 volt range. The outputs of the ADC1210 are buffered using 74C901 hex inverting TTL buffers and are then connected to the inputs of 74LS373 latches.

A schematic diagram of the A/D converter is shown in Figure 3.9. To start conversion, the START CONVERSION (SC*) pin is taken low by asserting ADCH1*, ADCH2*, or ADCH3*. This causes the SA register to reset synchronously on the next clock cycle low-to-high transition. Conversion begins on the next low-to-high transition of the clock pulse. After completing the conversion, the CC* signal is asserted. This output, together with a 74121 monostable multivibrator, generates the ADIRQ* signal. The timing diagram of the A/D operation is shown in Figure 3.10. Asserting ADREAD* latches the A/D output data to the MC68000 data bus.

The ADTACK* signal is generated using the circuit shown in Figure 3.11. There are two cases in which this signal is generated. The first case arises when the HOLD* or ADREAD* signals are asserted. These signals operate synchronously with the processor. Using the 74LS175 device, a delay corresponding to 4 clock cycles of the ECB 8MHZ CLOCK signal is incurred before ADTACK* is sent to USER DTACK*. The second case arises as a result of the asynchronous operation of the A/D converter and the processor. The ADTACK* signal is asserted only when the A/D converter SA register has been reset and the converter is about to start its conversion process. The timing diagram for this case is shown in Figure 3.12.

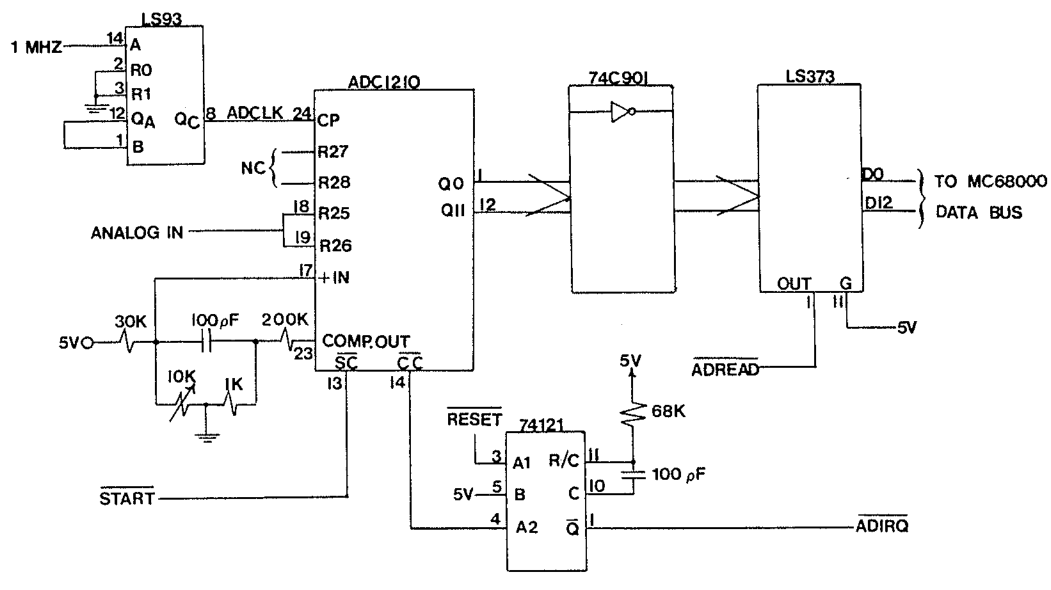


Figure 3.9: Schematic Diagram of ADC1210 A/D Converter

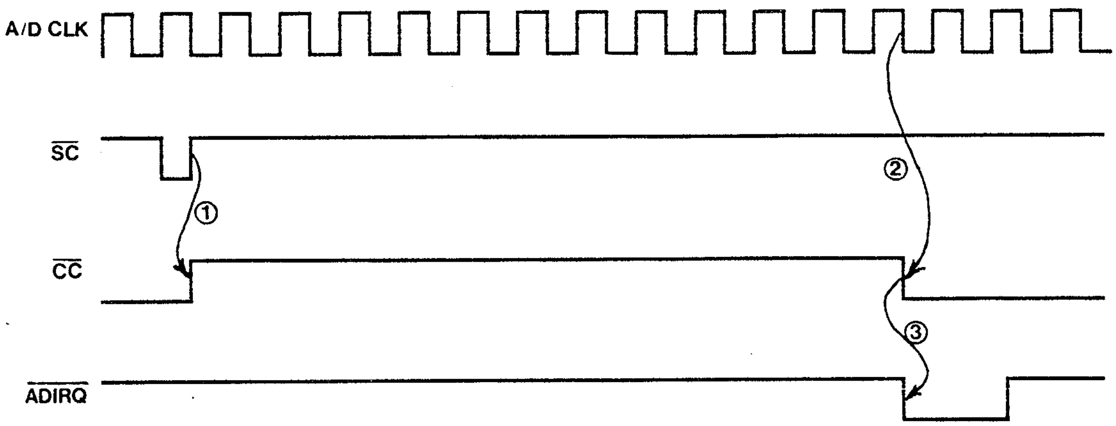


Figure 3.10: A/D Operational Timing Diagram

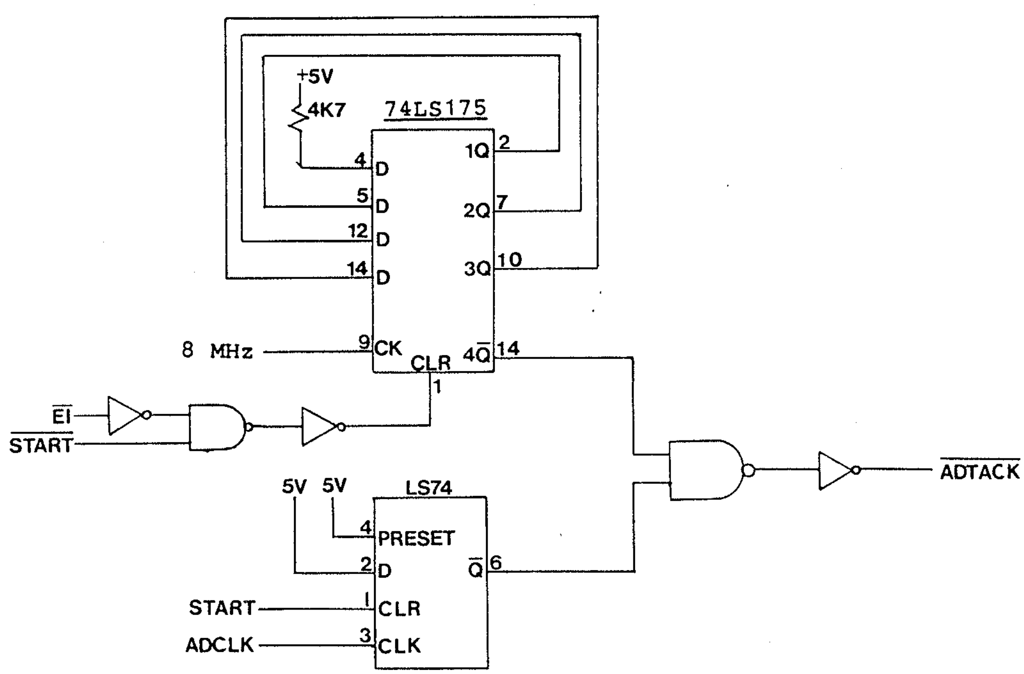


Figure 3.11: ADTACK* Signal Generation

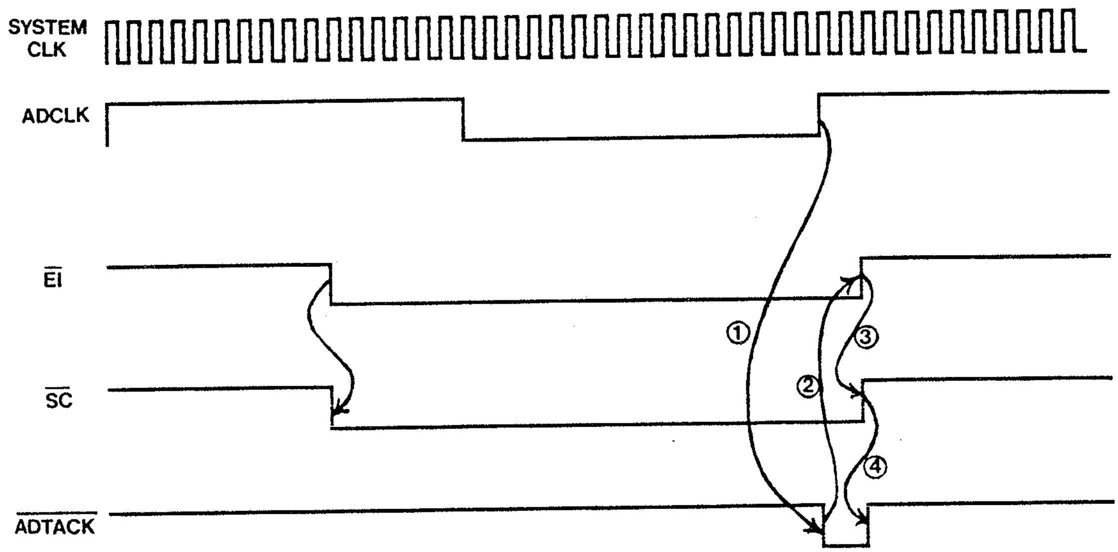


Figure 3.12: ADTACK* Timing Diagram

3.4 INTEL 8231A ARITHMETIC PROCESSING UNIT

The 4 MHz Intel 8231A (also Advanced Micro Devices Am9511) arithmetic processing unit (APU) performs fixed and floating point arithmetic and floating point trigonometric and mathematical operations. All transfers take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack and a command is issued to perform operations on the data in the stack. Results are then available on the stack or additional operations may be performed. Transfers to and from the APU are handled by the MC68000 over data lines D0-D7. These data lines are buffered using a 74LS245 device.

The APU is mapped into memory segment $050000-$05FFFF. ECB connection E3* and MC68000 address line A4 are used in the decoding scheme. The resulting addresses generated are $50001 for the operand entry address and $50011 for the command entry address.

A schematic diagram of the APU is shown in Figure 3.13. The C/D*, RD*, and WR* lines are used to determine the type of data transfer to be performed over the data bus. Table 3.7 describes the functions available and the corresponding signals on the C/D*, RD*, and WR* lines needed to provide these functions. A command may be issued after the required operands have been positioned on the stack. Upon completion of command execution, the END* output goes low. This signal is used to generate the APUIRQ* signal, informing the MC68000 that execution of the command is completed. The MC68000 sends an interrupt acknowledge signal to the APU by asserting APUEACK*. This signal resets the END* output on the APU.

The APUDTACK* signal is generated using a 74LS175 and 74LS74 device, as shown in Figure 3.14. The READY* output is used as a handshake signal during read or write transactions. A low on this output indicates the APU has not yet completed its information transfer with the MC68000. For read operations, after RD* and CS* go low, READY* will go low and remain low until the data bus contains valid data, then goes high. The READY* pulse low duration (TPPWR) can be a minimum of 0.9 μsec and a maximum of 3.05 μsec. For write operations, after WR* and CS* go low, READY* goes low for a very short duration. The READY* pulse low duration is a minimum of zero and a maximum of 50 nsec. This signal may not go low at all for very fast devices. The APUDTACK* circuitry handles these cases. The timing diagrams for the APU read and write operations are shown in Figure 3.15.

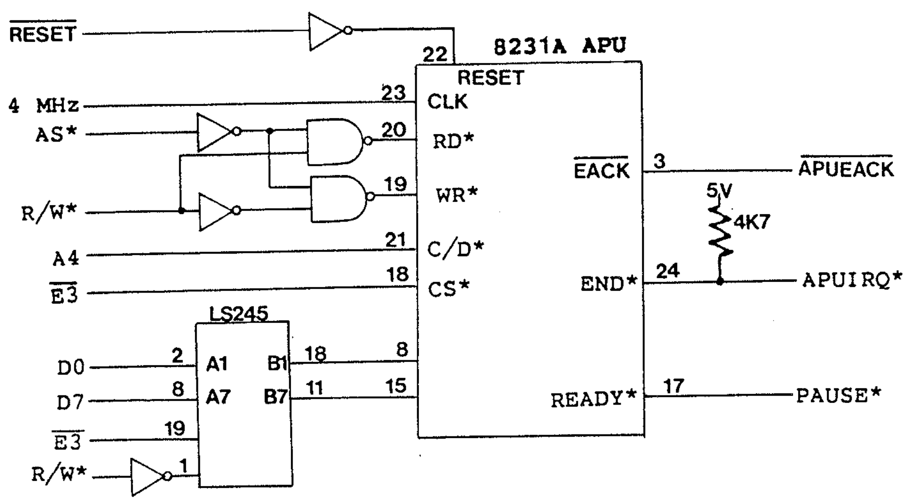


Figure 3.13: Schematic Diagram of INTEL 8231A APU

TABLE 3.7

APU Function Table

C/D*	RD*	WR*	Function
L	H	L	Push data byte onto stack
L	L	H	Pop data byte from stack
H	H	L	Enter command byte from data bus
H	L	H	Read status
X	L	L	Undefined

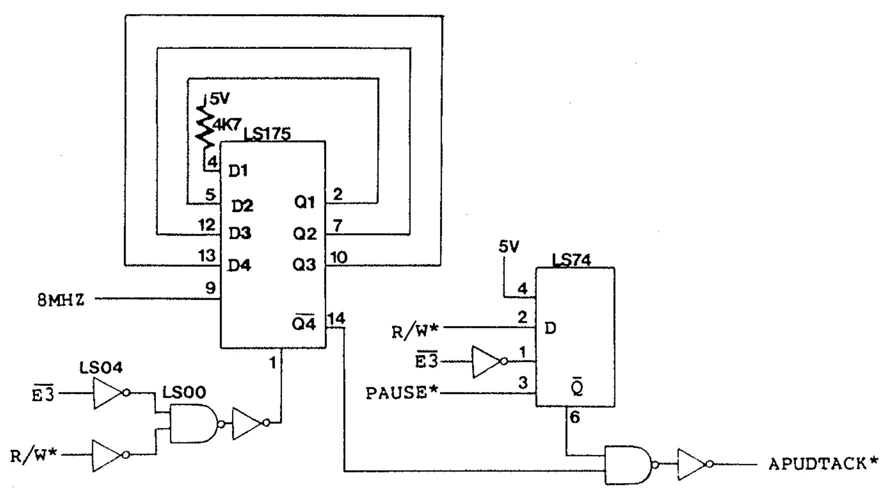
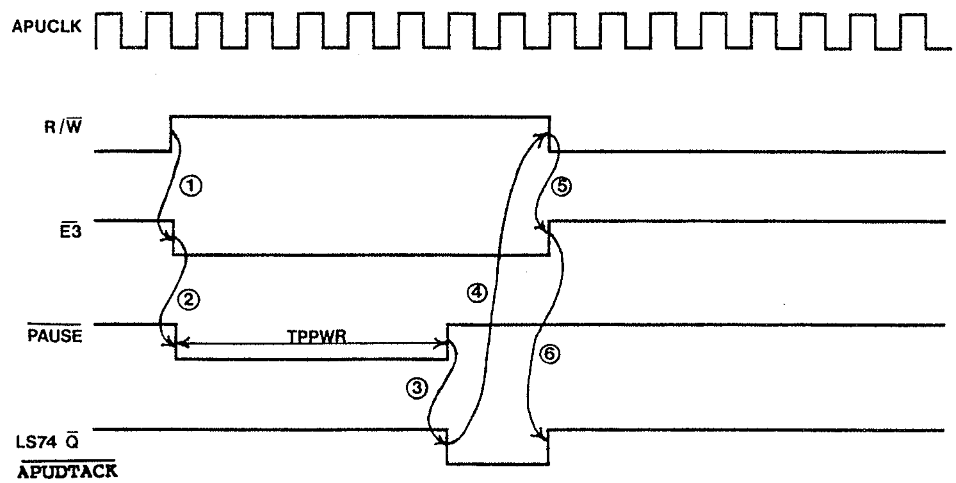


Figure 3.14: APUDTACK* Signal Generation

read:



write:

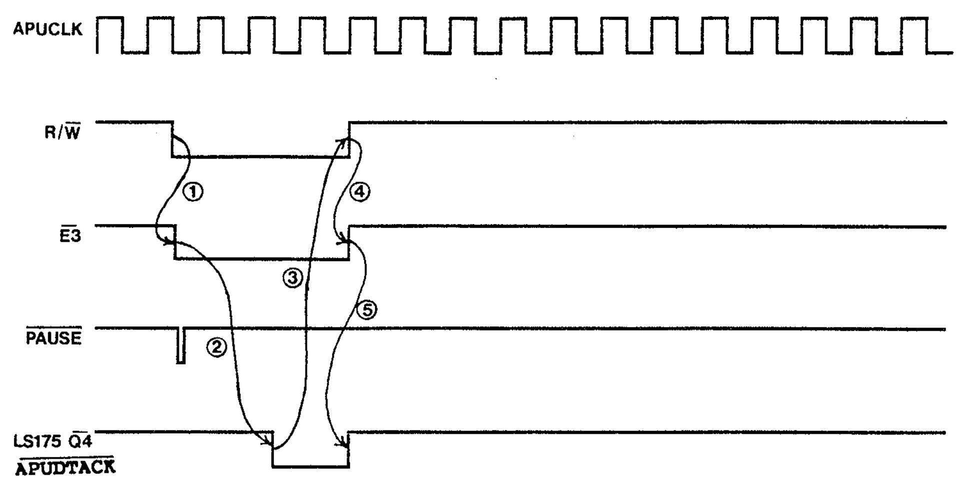


Figure 3.15: APUDTACK* Timing Diagrams

3.5 DIGITAL-TO-ANALOG CONVERSION CIRCUITRY

The digital-to-analog (D/A) converters are mapped in memory segment \$040000-\$04FFFF. Control of the D/A converters is provided with a 74LS138 3- to 8-line decoder. Connection E2* from the ECB and MC68000 address lines A1-A3 are used in the decoding scheme, as shown in Figure 3.16. Table 3.8 shows the D/A control signal address map.

Three National Semiconductor DAC1210 12-bit D/A converters are interfaced directly to the MC68000 data bus. These devices are configured to appear as one-word addresses in the I/O space. Conversion of a 12-bit digital word to an analog signal is performed by writing the data to D/A memory location DACHx (where x = 1, 2, or 3). The output of the D/A converter is a current (IOUT1); thus a current-to-voltage converter circuit using an MC741 operational amplifier is used to provide an output voltage in the range 0.0-5.0 volts. A schematic diagram of one of D/A converters as it is configured in this system is shown in Figure 3.17.

The DADTACK* signal is generated using the 74LS175 configuration similar to that shown in Figure 3.11. The E1* signal is replaced by E2* and a delay corresponding to 4 clock cycles of the ECB 8MHZ CLOCK signal is incurred before the DADTACK* signal is sent to the USER DTACK* circuitry.

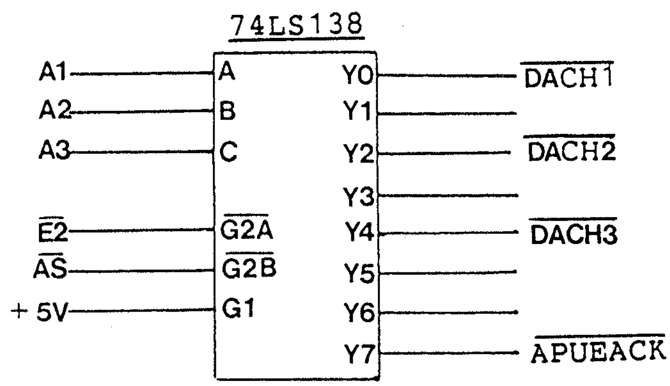


Figure 3.16: D/A Circuitry Decode Logic

TABLE 3.8

D/A Control Signal Address Map

Address	Function
$040000	Perform D/A conversion on channel 1 (DACH1)
$040002	Not used
$040004	Perform D/A conversion on channel 2 (DACH2)
$040006	Not Used
$040008	Perform D/A conversion on channel 3 (DACH3)
$04000A	Not used
$04000C	Not used
$04000E	APU Interrupt Acknowledge (APUEACK)

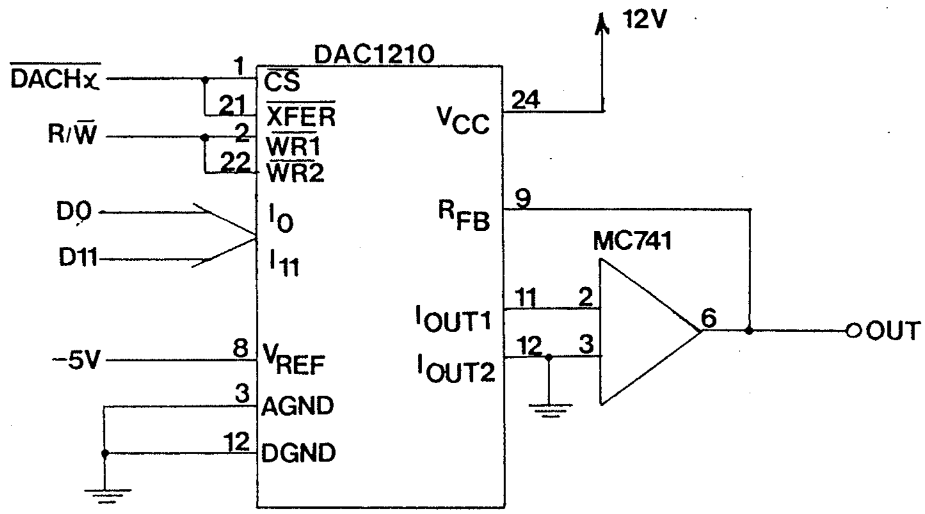


Figure 3.17: Schematic Diagram of DAC1210 D/A Converter

Chapter IV

SIGNAL PROCESSING TECHNIQUES AND SOFTWARE DESCRIPTION

4.1 INTRODUCTION

Frequently, murmurs or alterations in the heart sounds are the only definitive signs of some types of heart disease, appearing long before stress on the cardiovascular system is sufficient to produce other signs and symptoms [3]. Presently, auscultation is the primary test performed to assess the condition of the heart; however, this test is prone to interpreter variations and errors [22,23]. The primary aim in automated PCG analysis is the quantification of the subjective features of the signal. The important features of the PCG signal are the intensity and timing sequence of murmurs and sounds, and the location, frequency content, and envelope shape of murmurs, if present. Currently, there exists the need for a simple and efficient method to quantify the PCG signal into parameters aiding the detection of murmurs. Such a system would be extremely useful for screening purposes and routine diagnosis.

In this chapter, methods for the time and frequency domain analysis of the systolic and diastolic segments of the phonocardiogram (PCG) signal are presented. A technique for the segmentation of the PCG into systole and diastole is also discussed. The electrocardiogram (ECG) and the carotid pulse signals are used as timing references for the first and second heart sounds. Using a QRS-complex detection algorithm based on a smoothed difference of the ECG signal, the RR interval, corresponding to one cardiac cycle, can be established. The beginning of the cardiac cycle coincides with the start of the first heart sound (start of systole). A transformation based on a smoothed second difference of the carotid pulse is applied to locate the dicrotic notch in the carotid pulse. Using the location of the dicrotic notch, the beginning of the second heart sound (start of diastole) can be determined.

Upon segmenting the PCG into systole and diastole, quantification of the time and frequency domain characteristics is conducted. The energy curve and energy spectrum of each segment are computed and are then quantified using the concept of an Energy Distribution Coefficient (EDC).

The signal processing software is written in MC68000 assembly language using the MC68000 two-pass Cross Macro Assembler resident on the Amdahl 580/5850 computer system. This assembler provides access to the host computer's resources including the text editor and filing system. For a complete description of the assembler and an instruction set summary refer to the assembler reference manual [102].

For debugging purposes, the software can be downloaded from the host computer system into the Educational Computer Board (ECB) memory. The ECB operates under control of "TUTOR" firmware. TUTOR is the system monitor which controls communication with the terminal and host computer and provides debug capabilities, line assembly/disassembly, and I/O control. Upon completion of the debugging process, the software can be programmed into an erasable programmable read only memory (EPROM).

The signal processing software can be divided into the following major categories (Figure 4.1):

1. System Initialization
2. Analog-to-Digital Conversion
3. Detection of the QRS-Complex in the ECG
4. Detection of the Dicrotic Notch in the Carotid Pulse
5. Segmentation of the PCG into Systole and Diastole
6. Computation of the PCG Energy Curves
7. Computation of the PCG Energy Spectra
8. Computation of the Energy Distribution Coefficients

This chapter provides a detailed description of the signal analysis theory and techniques and the signal processing software. A program listing is included in Appendix B. In this discussion numbers in hexadecimal form are represented with a preceding '$'.

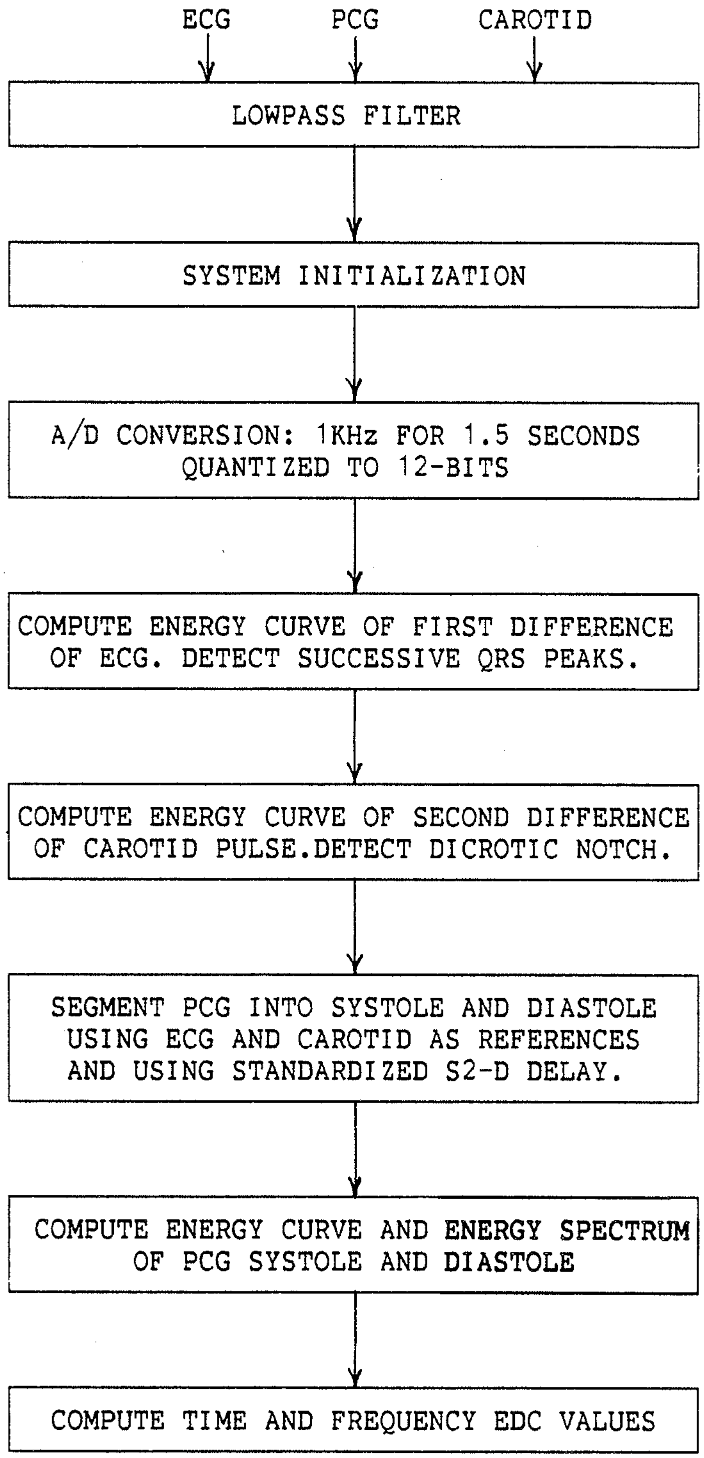


Figure 4.1: Signal Processing Flowchart

4.2 SYSTEM INITIALIZATION

The MC68000 microprocessor contains a 16-bit status register (Figure 4.2) which contains the interrupt mask as well as the following condition codes: extend, negative, zero, overflow, and carry. Additional status bits indicate that the processor is in a trace mode and in one of two states of privilege: the supervisor state or the user state. The initialization procedure begins with selecting the supervisor state for the MC68000. The supervisor state is the higher state of privilege in which all instructions can be executed. The interrupt mask of the status register is initialized to allow level 4 interrupts or higher.

Since the processor is in the supervisor state, the 32-bit supervisor stack pointer is set to a valid RAM address. The seven 32-bit address registers and eight 32-bit data registers are initialized as necessary throughout the program.

Initialization of the starting address of the A/D and APU interrupt service routines (ISR) is also required. Interrupt vectors are memory locations from which the processor fetches the address of a routine which will handle the interrupt. A vector number is an 8-bit number which, when multiplied by four, gives the address of an interrupt vector. When the M6800 IRQ* signal is asserted a level 4 interrupt acknowl-

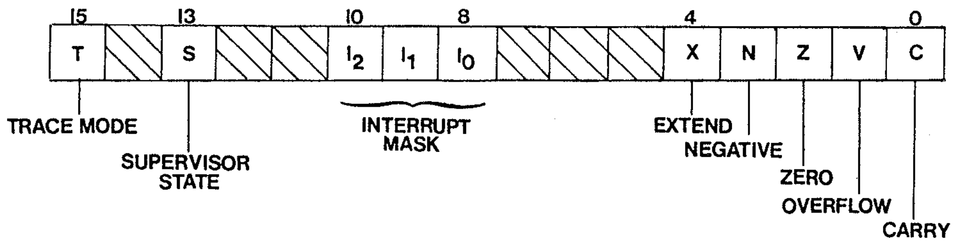


Figure 4.2: MC68000 Status Register

edge cycle from the MC68000 causes an autovectored response with the vector number equal to 28. Autovectoring is used when the interrupting device can not provide the processor with the vector number. The address of the A/D and the APU interrupt service routines is thus stored in the interrupt vector at address $000070.

4.3 ANALOG-TO-DIGITAL CONVERSION

The PCG, ECG, and carotid pulse signals are sampled at a rate of 1024 Hz. The sampling period is controlled using the timer in the PI/T. The timer is initialized by writing the sampling period to the Counter Preload Register (CPR). The timer is set up to load the Counter Register (CR) from the CPR and to begin decrementing the CR. At the start of the sampling period, the S/H device hold state is initiated by performing a 'dummy' write to address HOLD. During each sampling period, data from the three analog channels are digitized and stored in RAM. A flowchart of the A/D process is shown in Figure 4.3. The following events are carried out for each analog channel:

1. A 'dummy' write to address ADCH1, ADCH2, or ADCH3 is conducted to connect the analog multiplexer input to the output and to start the A/D converter.
2. The processor stops fetching and executing instructions until the CONVERSION COMPLETE* signal is received from the A/D converter (via the M6800 IRQ* level 4 interrupt).
3. The A/D ISR address is fetched from the level 4 interrupt vector and the program enters the ISR. The digitized data is read from address ADREAD and stored in the system memory.

After the three analog signals have been digitized, the processor waits until the sampling period is over (CR equals zero). Upon reaching zero, the CR is loaded from the CPR and resumes counting and the sampling process continues. Approximately 1.5 seconds of data are acquired and stored in RAM, upon which the timer is disabled.

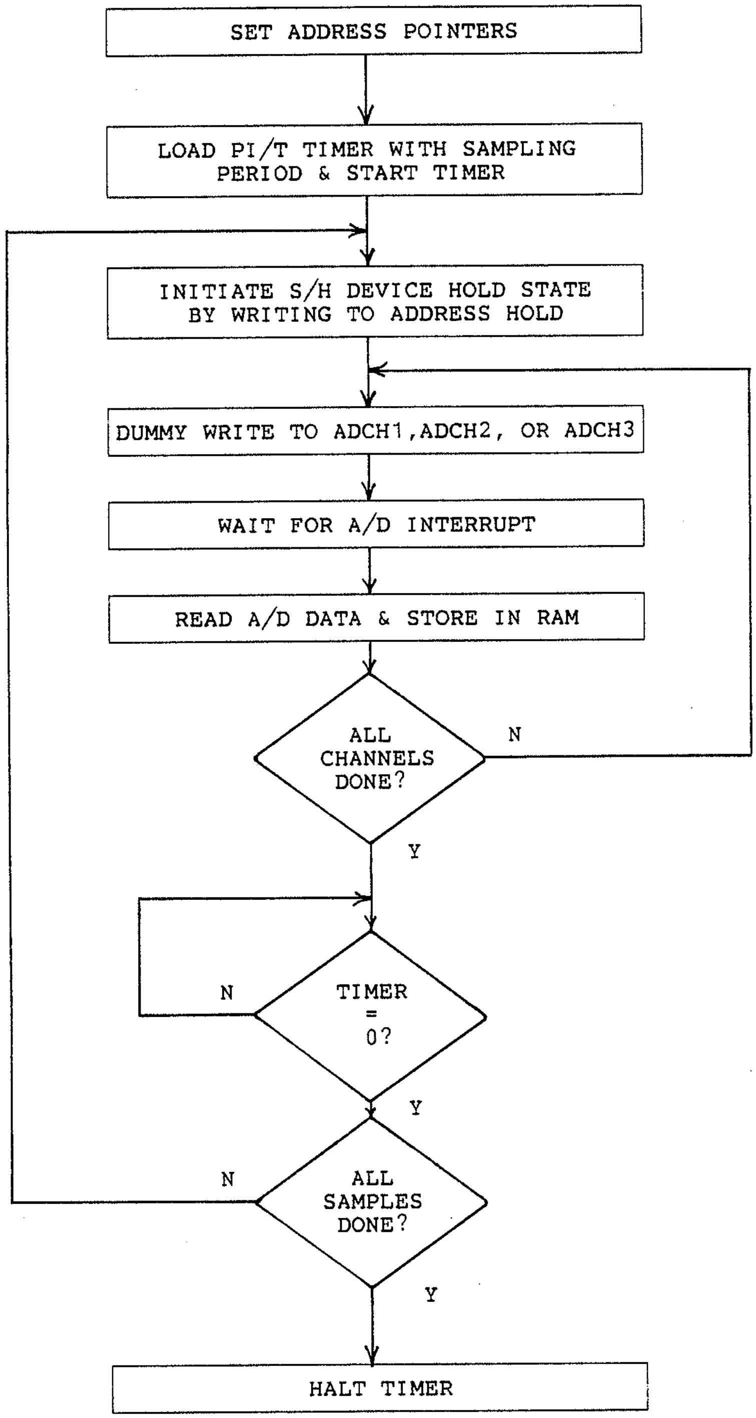


Figure 4.3: Analog-to-Digital Conversion Flowchart

4.4 DETECTION OF THE QRS-COMPLEX IN THE ECG

The digitized ECG signal is first downsampled to an effective sampling rate of 256 Hz by taking every fourth sample. A QRS-complex detection routine based on a smoothed difference of the digitized ECG signal is used to determine the RR interval [88,89]. With $x(n)$ representing the digitized ECG signal, the following transformation is applied to compute the smoothed energy curve of the first difference of the ECG signal:

$$g(n) = \sum_{k=1}^{M} [x(n-k+1) - x(n-k)]^2 w(k) \qquad (4.1)$$

Here $w(k)$ is a weighting sequence or window which selects a segment of $x(n)$, M is the number of samples in the window, and n is the running index of the signal and its transform. The purpose of the window is to attach lower weight to samples which occurred further back in time. Thus $w(k)$ tends to zero monotonically as k increases. We define $w(k) = (M-k+1)$, where $k = 1,2...M$. This process, in effect, convolves the ECG difference signal with a decreasing weighting sequence.

The ECG transform $g(n)$ yields a single positive peak for each cardiac cycle with maximum value occurring near the end of the QRS-complex. An ECG signal and its corresponding transform $g(n)$ with $M=16$ are shown in Figure 4.4. In this figure note that the ECG signal is corrupted with low amplitude 60 Hz noise. This noise was picked up during the recording procedures. The low level noise, however, does not affect g(n). Since the amplitude of the noise is minimal in comparison with the amplitude of the QRS-complex, the noise is effectively smoothed out during the convolution process.

A simple peak detecting routine is used to detect successive QRS end points (Q1,Q2) from g(n). The ECG transform is scanned and a threshold is set at half the maximum transform value. The transform data is scanned a second time to search for the QRS end points. A sample in g(n) is considered a QRS end point peak if the sample value at that point is greater than the threshold and is greater than the ten previous sample values and the ten following sample values. This search is continued until two peaks are found, representing the start and end of one cardiac cycle.

Figure 4.5 shows the carotid pulse and the ECG transform g(n). After detecting two successive peaks in g(n), denoted Q1 and Q2, point C1 is identified on the carotid pulse. This point corresponds to the start of the cardiac cycle. The dicrotic notch is searched for in a region starting at C1.

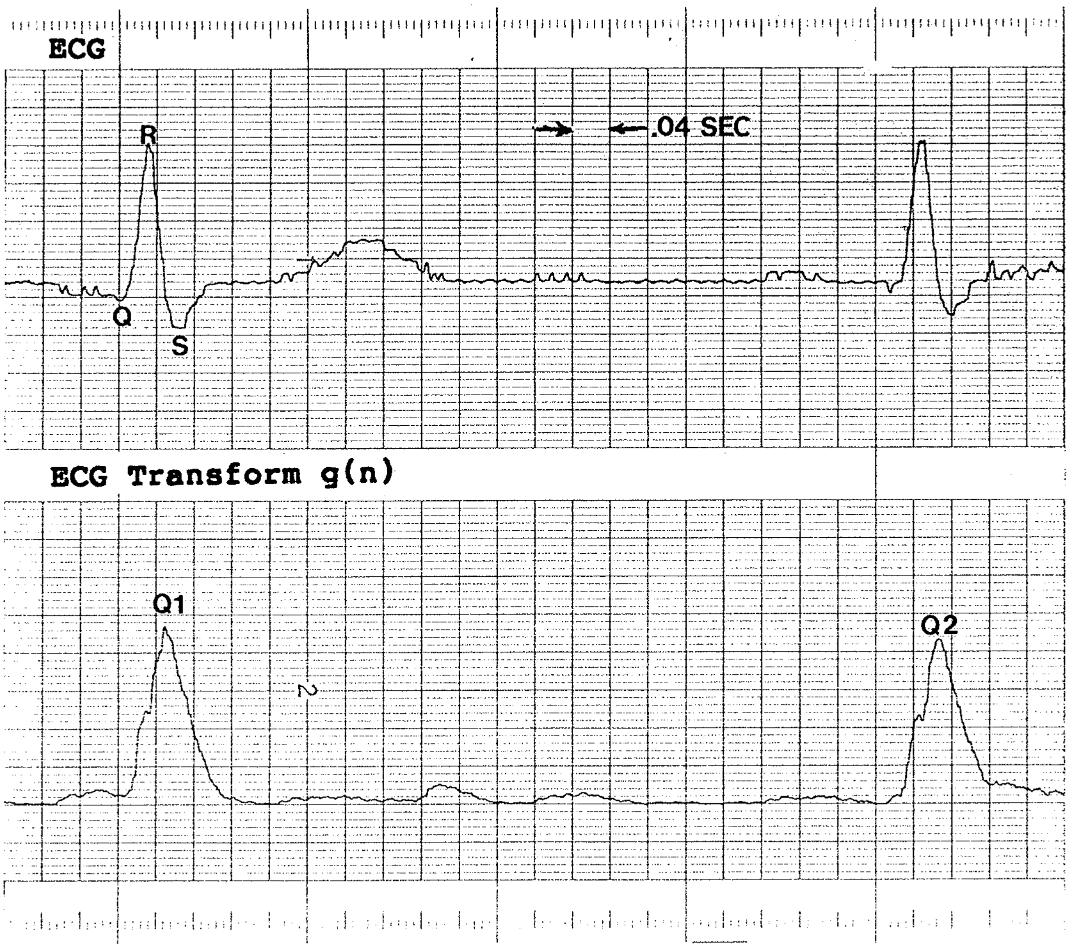


Figure 4.4: ECG and ECG Transform g(n)

Note that the ECG transformation g(n) yields a single positive peak for each cardiac cycle with maximum value occurring near the end of the QRS-complex.

The 32-bit g(n) sample values have been truncated to 12-bits for display purposes.

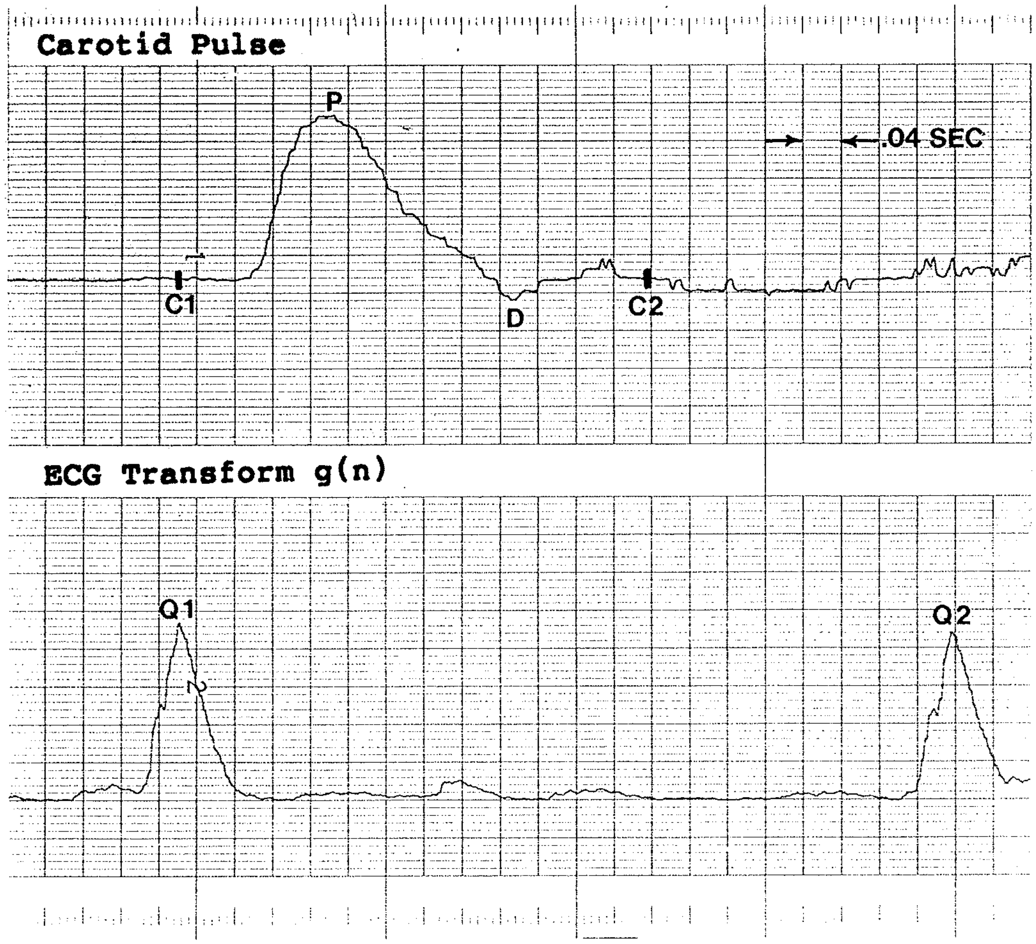


Figure 4.5: Carotid Pulse and ECG Transform g(n)

Point C1 on the carotid pulse corresponds to the start of the cardiac cycle. C2 is the point 500 msec after C1.

The 32-bit g(n) sample values have been truncated to 12-bits for display purposes.

4.5 DETECTION OF THE DICROTIC NOTCH IN THE CAROTID PULSE

The carotid pulse can be characterized by measuring the important intervals of the waveform. The pre-ejection period (PEP) is the period from the beginning of the QRS-complex of the ECG to the onset of the carotid upstroke as indicated in Figure 2.7. The following equation may be used to approximate the true PEP interval [25]:

$$PEPC = PEP + 0.4(HR) \tag{4.2}$$

Here PEPC is the rate-corrected pre-ejection period (normally 131$\pm$13 msec), PEP is the actual pre-ejection period, and HR is the heart rate in beats per minute (bpm). It should be noted that this is only an approximate relation which depends on, among other factors, age, sex, and physiological condition of the heart.

Another indirect measurement is the ejection time (ET), which is the interval from the onset of carotid upstroke to the dicrotic notch, as indicated in Figure 2.7. In the normal resting individual, this interval is dependent upon heart rate. Using the following equation, the true ejection time (ET) can be approximated by [25]:

$$ETC = ET + 1.6(HR) \tag{4.3}$$

Here ETC is the rate-corrected ejection time, ET is the actual ejection time, and HR is the heart rate in bpm. The normal value for ETC is 395$\pm$13 msec for males and 415$\pm$11 msec for females. It should be noted that although these equations yield only approximate time intervals, they can be used to localize the dicrotic notch in a general area.

Using equation (4.2), the maximum interval between the end of the QRS-complex and the onset of carotid upstroke can be determined. Using a maximum PEPC of 144 msec and a minimum heart rate of 60 bpm, the PEP is 120 msec. It should be noted that equation (4.1) yields a peak near the end of the QRS-complex, therefore the interval between the S-wave of the ECG and the onset of carotid upstroke is 90 msec (using a minimum QRS width of 30 msec). The ejection time of the carotid pulse is the time interval from the onset of upstroke to the dicrotic notch. Equation (4.3) can be used to approximate the region in which the dicrotic notch will fall. Using a minimum heart rate of 60 bpm and a maximum ETC of 425 msec, the ET is 325 msec. Using the PEP and ET intervals, the maximum interval between the QRS-complex and the dicrotic notch is 380 msec. To take into account variational factors, such as abnormal PEP or ET intervals and formula approximations, it was decided to search for the dicrotic notch in an area between C1 and a point 500 msec after C1. This point is designated C2 on Figure 4.5.

The digitized carotid pulse is first downsampled to 256 Hz. The dicrotic notch is defined as the point of minimum pressure occurring after the percussion wave, in the region of the maximum second derivative. A least squares estimate of the second derivative [98] is obtained from

$$p(n) = [2y(n-2) - y(n-1) - 2y(n) - y(n+1) + 2y(n+2)] \quad (4.4)$$

where $y(n)$ represents the digitized carotid pulse waveform, and $p(n)$ is the second derivative evaluated at the nth data point. The following transformation is then applied to obtain a smoothed energy curve of the second derivative of the digitized carotid pulse:

$$s(n) = \sum_{k=1}^{M} p^2(n-k)w(k) \quad (4.5)$$

Here $w(k)$ is a window which selects a segment of $p(n)$, M is the number of samples in the window and n is the running index of the signal and its transform. The window $w(k)$ was selected such that $w(k) = (M-k+1)$, as for the ECG.

This transformation yields two positive peaks with local maxima occurring at points corresponding to the onset of carotid upstroke and the dicrotic notch. The carotid pulse and its corresponding transform with M=16 are shown in Figure 4.6. The first peak (T1) represents the onset of upstroke. After T1 is detected, the search for T2, the peak corresponding to the dicrotic notch, is undertaken. A peak detecting routine, similar to that presented in the previous section, is used to locate both peaks. To locate the dicrotic notch, the local minimum which occurs on the carotid pulse within a $\pm$ 20 msec region of the maximum second derivative (T2) is determined.

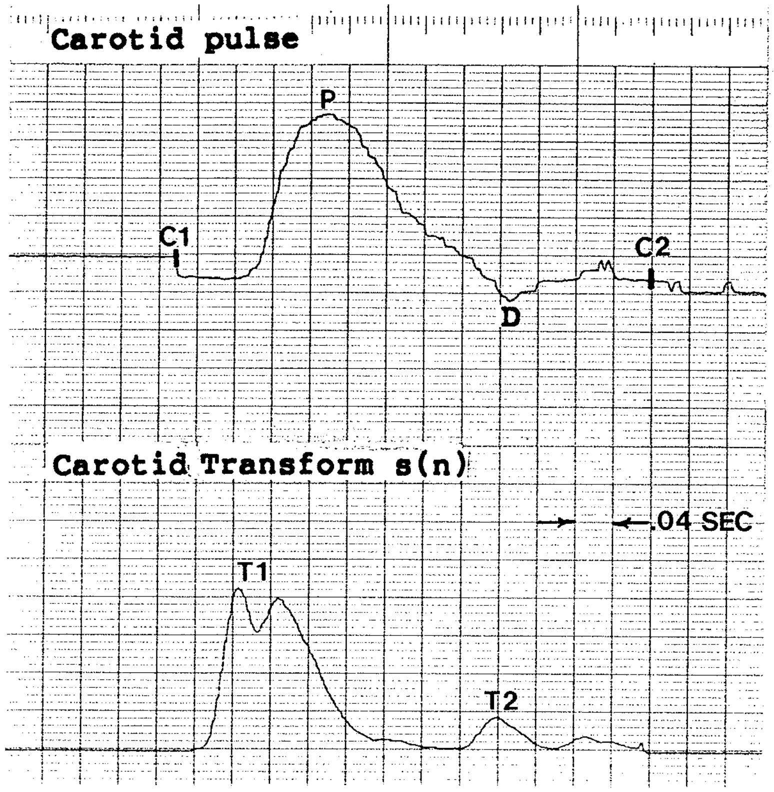


Figure 4.6: Carotid Pulse and Transform s(n)

The carotid transform s(n) results in two relative maxima. Peak T1 corresponds to the onset of carotid upstroke and peak T2 corresponds to the dicrotic notch.

The 32-bit s(n) sample values have been truncated to 12-bits for display purposes.

4.6 PCG SIGNAL PROCESSING

4.6.1 Segmentation of PCG into Systole and Diastole

The ECG transform g(n) is used to identify the start and end of one PCG cycle. Figure 4.7 shows the PCG signal and the corresponding g(n). The peaks in g(n) provide a sharp reference which can be used to determine the start of the first heart sound and thus the start of the cardiac cycle. The PCG cycle ends at the beginning of the next first heart sound. The start and end points of one cardiac cycle are labelled P1 and P3, respectively, on the PCG signal.

The dicrotic notch in the carotid pulse is used to identify the start of the second heart sound (S2). With the start of S2 determined, segmentation of the PCG into systole and diastole is possible. Figure 4.8 shows a PCG signal and the corresponding carotid transform s(n). As can be seen in the figure, there is a time delay between the start of S2 and the dicrotic notch (T2 on carotid transform - this interval will be referred to as the S2-D delay). This time delay must be properly determined in order to correctly identify the start of S2.

The S2-D delay is dependent upon the distance of the recording site from the heart and the pulse-wave velocity. Since the recording site of the carotid pulse is kept relatively constant, it was decided to use a measurement based on a standardized S2-D interval in order to correctly locate the start of S2 [103,104]. For the cases tested the S2-D delay was initially measured from a strip chart recording of the signals. This delay is reflected in column 1 of Table 4.1. Column 2 contains the heart rate of the patient. The range of the S2-D delay is quite small, ranging from a minimum of 35 msec to a maximum of 54 msec. For the signals analyzed in this study the mean delay was found to be 42.6 msec with a standard deviation of 5.0 msec. The mean delay and the standard deviation of the S2-D delay are used to compute the standardized S2-D interval.

The mean S2-D delay is subtracted from the computed location of the dicrotic notch. To prevent the likelihood that any portion of S2 is included in the systolic segment, a value equal to two standard deviations of the mean delay (in this work 10 msec) is further subtracted. This point, which identifies the start of S2, is labelled P2 on the PCG signal (Figure 4.8).

To summarize, the standardized S2-D interval is equal to the mean delay plus two standard deviations of the mean delay. In this work the interval is 53 msec. This value is subtracted from the computed location of the dicrotic notch to give the location of the start of the second heart sound. Figure 4.9 shows a PCG signal separated into its systolic and diastolic segments. Qualitatively, the PCG segmentation is quite accurate, as can be seen in this figure. Minor problems which may result in this method and an analysis of the PCG quantification error resulting from this procedure are discussed in the following chapter.

We believe this procedure to be quite accurate for quantifying the time and frequency domain characteristics of the systolic and diastolic segments of the PCG. This method is computationally simpler than other methods presented for PCG segmentation and analysis of the first and second heart sounds [40,42,45,54,55,57,58,67]. The minimal computation time makes it very useful for online analysis using a dedicated microcomputer system. In most reported cases, the exact location of the start of S2 is not used; rather, a window which selects the approximate location of S2 is used in the analysis [40,42,45,67]. Other reported methods involving linear prediction coding [57,58] are not suited for this type of microcomputer system, while statistical methods [57,58] require many cardiac cycles to perform PCG segmentation and are thus time consuming.

The procedure proposed in this work requires a third signal - the carotid pulse, which is a useful diagnostic tool recommended to be recorded with the PCG and ECG signals [105]. The use of the standardized S2-D interval simplifies the problem of accurate S2 location. This procedure has been successively implemented on the PCG analysis system and is used to segment the PCG into systole and diastole in order that time and frequency domain analysis can be performed on both PCG segments.

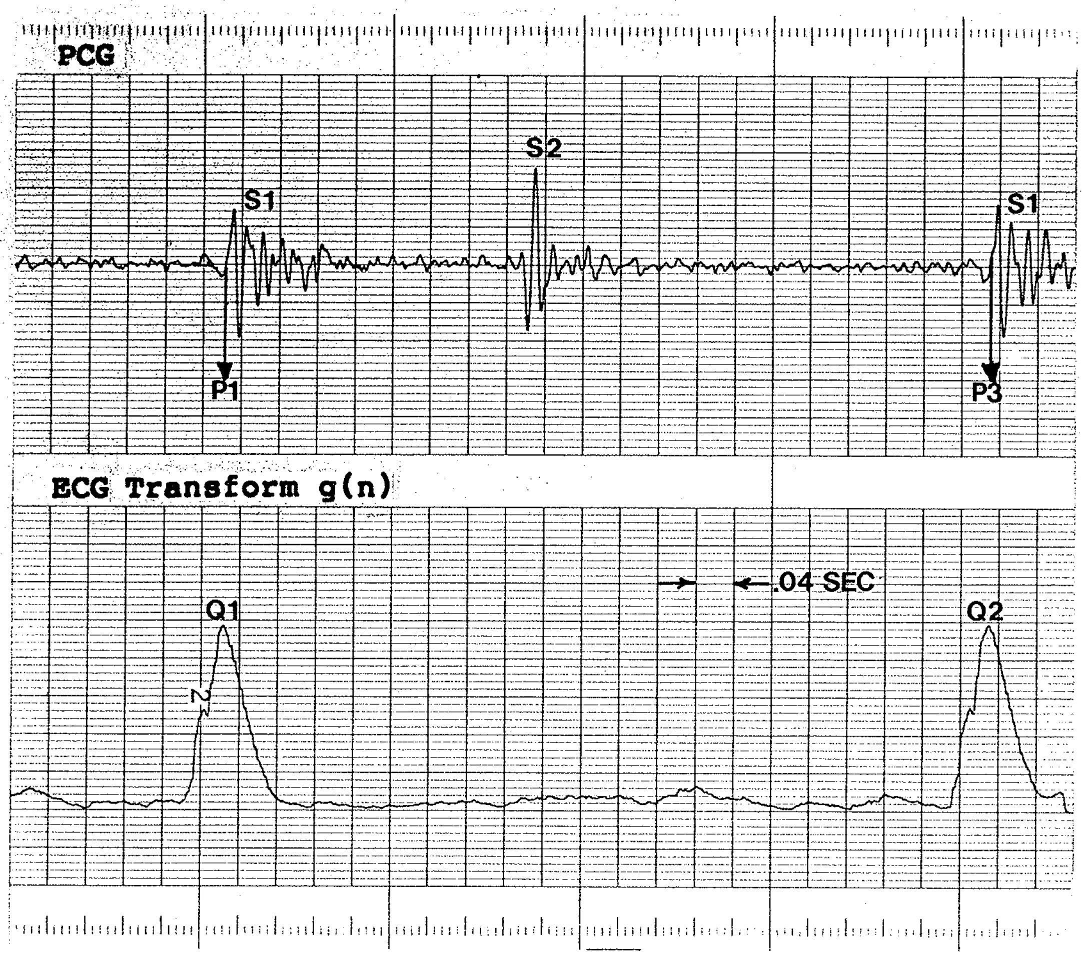


Figure 4.7: PCG signal and Corresponding ECG Transform g(n)

The 32-bit g(n) sample values have been truncated to 12-bits for display purposes.

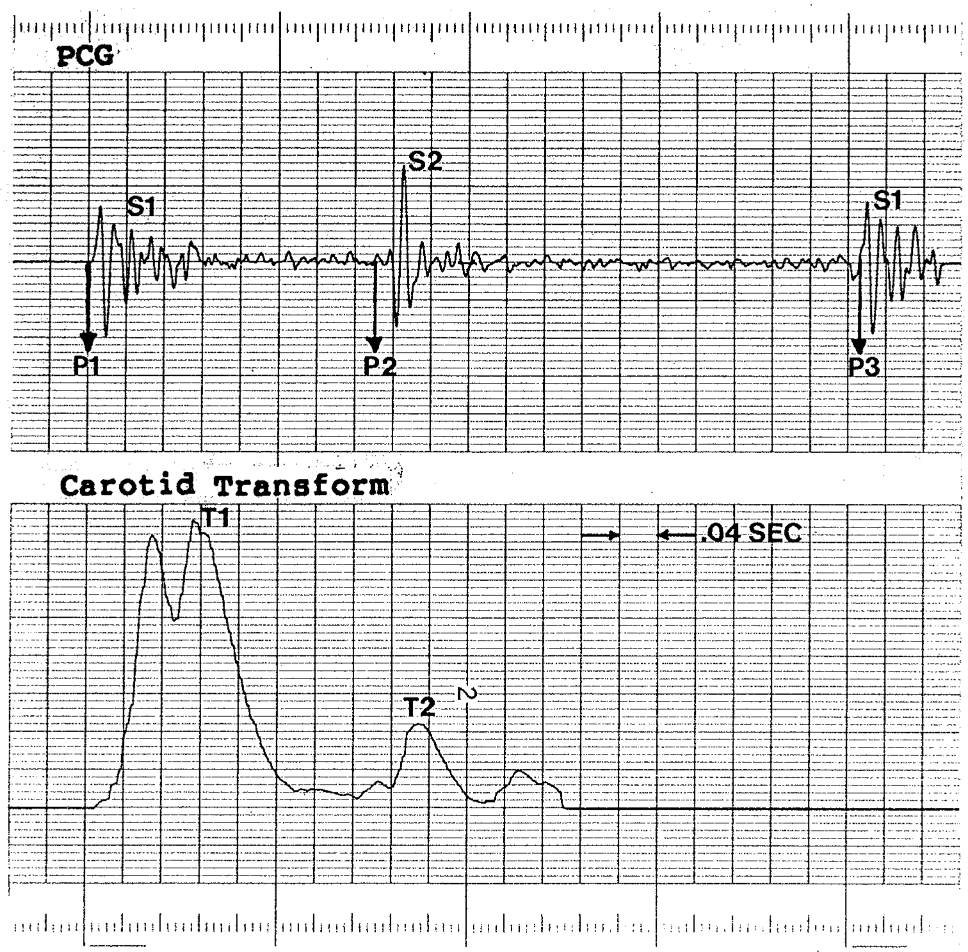


Figure 4.8: PCG signal and Corresponding Carotid Transform s(n)

The 32-bit s(n) sample values have been truncated to 12-bits for display purposes.

TABLE 4.1

Delay Between Second Heart Sound and Dicrotic Notch

Patient	S2-D Delay (msec)	Heartrate (bpm)
A	45	61
B	46	73
C	44	70
1a	42	77
1b	43	76
1c	44	76
2a	53	75
2b	53	75
2c	54	72
3a	41	87
3b	39	85
3c	38	83
4a	43	111
4b	44	117
5	38	82
6	39	84
7a	40	83
7b	44	83
8a	52	62
8b	50	60
9a	41	92
9b	39	94
10	41	77
11a	47	82
11b	45	93
13	47	82
14a	46	80
14b	48	80
15b	42	74
15c	44	78
16a	35	130
17a	44	76
18a	39	90
20	35	67
21	36	82
22a	35	45
22b	41	51
22c	41	51
24a	35	54
24b	41	58

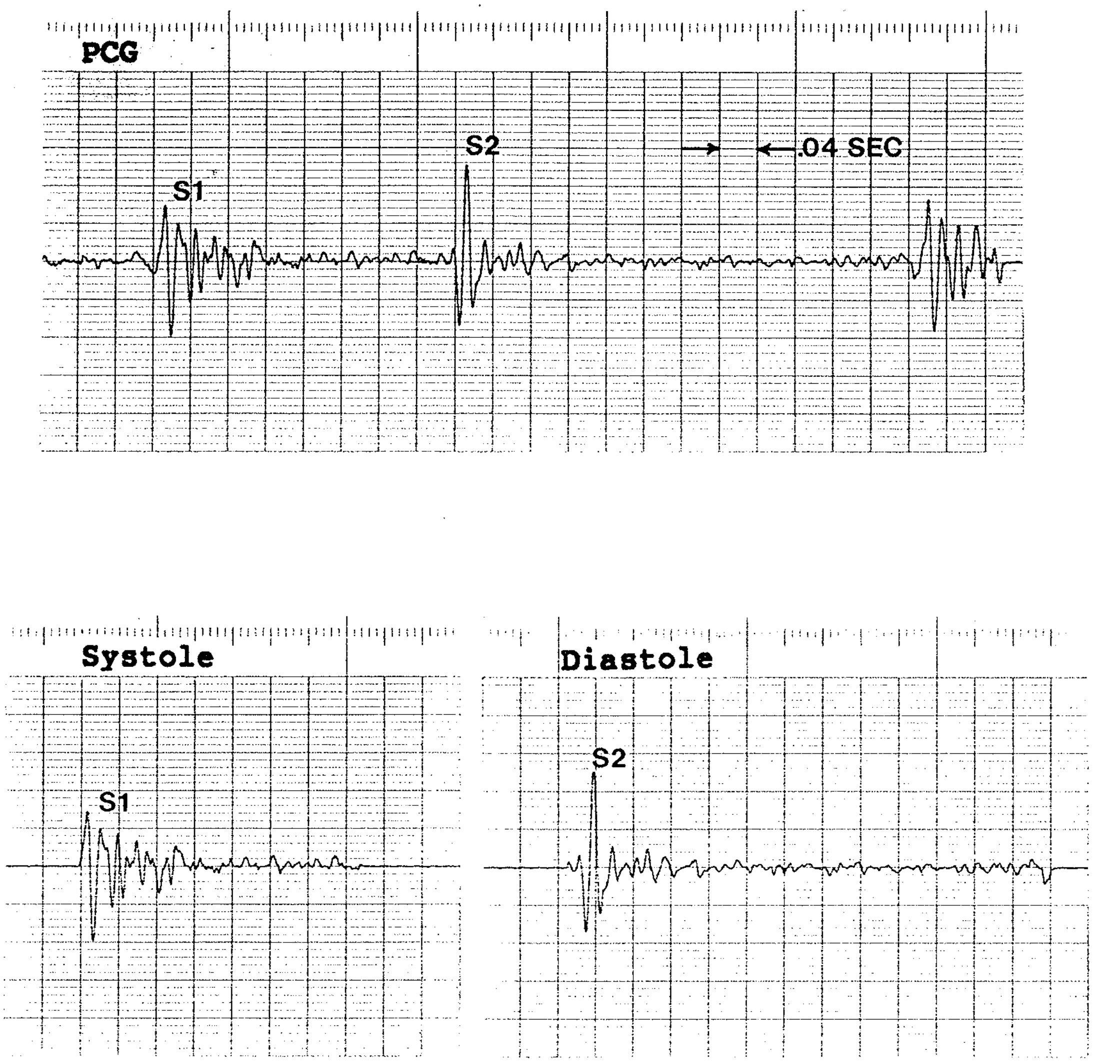


Figure 4.9: PCG Segmented into Systole and Diastole

4.6.2 PCG Energy Curve

The pattern of energy distribution of the PCG signal is very important in the diagnosis of valvular and septal defects, which cause murmurs of characteristic envelope shapes in particular locations of the cardiac cycle. This work employs the method of computing energy covered by a moving window, which is commonly used in speech signal processing [106].

The energy of a signal is an easily extractable parameter. In the case of a real discrete time signal x(n), the energy is defined in general as [106]

$$E = \sum_{n=-\infty}^{\infty} x^2(n) \tag{4.6}$$

For nonstationary signals such as the PCG, a time-varying energy calculation can be used as follows:

$$E(n) = \sum_{k=1}^{M} x^2(n-k)w(k) \tag{4.7}$$

where w(k) is a decreasing weighting sequence which selects a segment of x(n) and M is the number of samples in the window. The sequence E(n) displays the time-varying energy characteristics of the signal x(n). The selection of a suitable window w(k) and appropriate window width M is very important in obtaining useful energy measurements. The window w(k) is taken as a decreasing function so that PCG samples which occurred further back in time will have lower weight attached with them. The choice of M depends on the nature of the signal and the sampling rate used. If M is too small there will be many ripples in E(n) making it appear noisy; however, if M is too large E(n) will become featureless [106]. With a sampling rate of 1024 Hz, M is chosen to be 32 and the window function is defined as $w(k) = M+1-k$.

This method is used to obtain the time-varying amplitude characteristics of the systolic and diastolic segments of the PCG signal for one cardiac cycle. The significance of E(n) is that it provides a good measure for separating the heart sounds and murmurs from the silent periods during the cardiac cycle. Thus, the presence of murmurs is indicated when significant energy appears after the first heart sound in the systolic segment or after the second heart sound in the diastolic segment.

4.6.3 PCG Energy Spectrum

The frequency content of the PCG is one of the easiest features that can be assessed by auscultation. It is also an important factor in the detection of murmurs. A normal PCG lies in the 0 - 200 Hz frequency band. The frequency content of murmurs may be as high as 600 Hz; however the frequency content depends upon the pressure gradient across the defect, which depends upon the extent of the defect. The energy spectrum of the signal, which is the square of the Fourier transform, gives useful information by providing the distribution of energy versus frequency.

Physically, the Fourier transform X(k) represents the distribution of signal strength with frequency. The fast Fourier transform (FFT) is a method for computing the discrete Fourier transform (DFT). With x(n) representing the signal, the DFT is defined by [107]

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk} \tag{4.8}$$

for k = 0,1,...,N-1. The term X(k) is the kth coefficient of the DFT and x(n) is the nth sample of the time series, consisting of N samples. The term $W_N = \exp(-j2\pi/N)$. The inverse discrete Fourier transform (IDFT) is given by

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, W_N^{-nk} \tag{4.9}$$

A typical FORTRAN source code of a radix-2 decimation in time FFT is shown in Figure 4.10. The FFT subroutine implemented in MC68000 assembly language is based on this algorithm [108]. To facilitate understanding of the assembly language version, each FORTRAN statement is reproduced in the comment field of the corresponding block of MC68000 instructions (See Appendix B).

The first part of the program, which is equivalent to the DO 3 loop, performs the in-place bit reversal shuffling of the input vector. The second part, equivalent to the nested DO loops, performs the actual FFT computations. Loop DO 4 I=J,N,LE evaluates the butterflies which have the same coefficient W_N^k in a given stage. Loop DO 5 I,LE1 computes and keeps track of which coefficient W_N^k is being evaluated in a given stage, while loop DO 5 L=1,LN runs through all M stages.

The 8231A APU performs 16- and 32-bit fixed-point operations, as well as 32-bit floating-point arithmetic. In addition to the basic arithmetic operations (add, subtract, multiply, divide) the APU also evaluates logarithmic and trigonometric functions. The APU commands operate on operands located at the top of the APU stack (TOS) and next on stack (NOS) and results are returned to the stack at NOS and then popped to TOS. Operands are entered into the stack least significant byte first and most significant byte last by writing to the APU operand entry address APUOPER. The APU stack can accommodate four floating-point quantities. After the operands are positioned on the stack, a command can be issued by writing to the APU command entry address APUCOM. An interrupt request from the APU signals the current command execution is completed. At this time, the result from an operation can be read from TOS or a new command can be given.

The time required to compute a 1024-point FFT is approximately 4.3 seconds. All calculations are done in floating-point arithmetic to eliminate scaling and rounding of values, and to provide a greater dynamic range. Figures 4.11 and 4.12 show examples of typical time signals and their respective transforms computed using the PCG analysis system.

Since the digitized PCG segment x(n) is of finite length, the data is multiplied by a Hamming window w(n) to improve the spectral quality of the output. This process convolves the Fourier transforms of x(n) and w(n). The Hamming window equation is as follows [109]:

$$w(n) = \begin{cases} .54 + (.46)\cos(2\pi n/N), & |n| < (N-1)/2 \\ 0 & , \text{otherwise} \end{cases} \qquad (4.10)$$

Here N is the number of samples in the segment.

The FFT routine is used to compute the Fourier transform of the systolic and diastolic segments of the PCG signal over one cardiac cycle. The energy spectrum, which is the square of the Fourier transform, is computed for both segments to obtain the distribution of energy versus the frequency. This method is used to obtain the frequency domain characteristics of the PCG for one cardiac cycle. An indication of the presence of high frequency murmurs is obtained when the energy spectrum displays significant energy beyond 200 Hz.

```
      SUBROUTINE FFT(F,LN)
      COMPLEX F(1024),U,W,T,CMPLX
      PI=3.141593
      N=2**LN
C
C   START BIT REVERSAL
C
      NV2=N/2
      NM1=N-1
      J=1
      DO 3 I = 1,NM1
      IF (I .GE. J) GO TO 1
      T = F(J)
      F(J) = F(I)
      F(I) = T
 1    K=NV2
 2    IF (K .GE. J) GO TO 3
      J = J-K
      K=K/2
      GO TO 2
 3    J=J+K
C
C   FFT COMPUTATION
C
      DO 5 L=1,LN
      LE=2**L
      LE1=LE/2
      U=(1.0,0.0)
      W=CMPLX(COS(PI/LE1),-SIN(PI/LE1))
      DO 5 J=1,LE1
      DO 4 I=J,N,LE
      IP=I+LE1
      T=F(IP)*U
      F(IP)=F(I)-T
 4    F(I)=F(I)+T
 5    U=U*W
      RETURN
      END
```

Figure 4.10: FFT FORTRAN Source Code

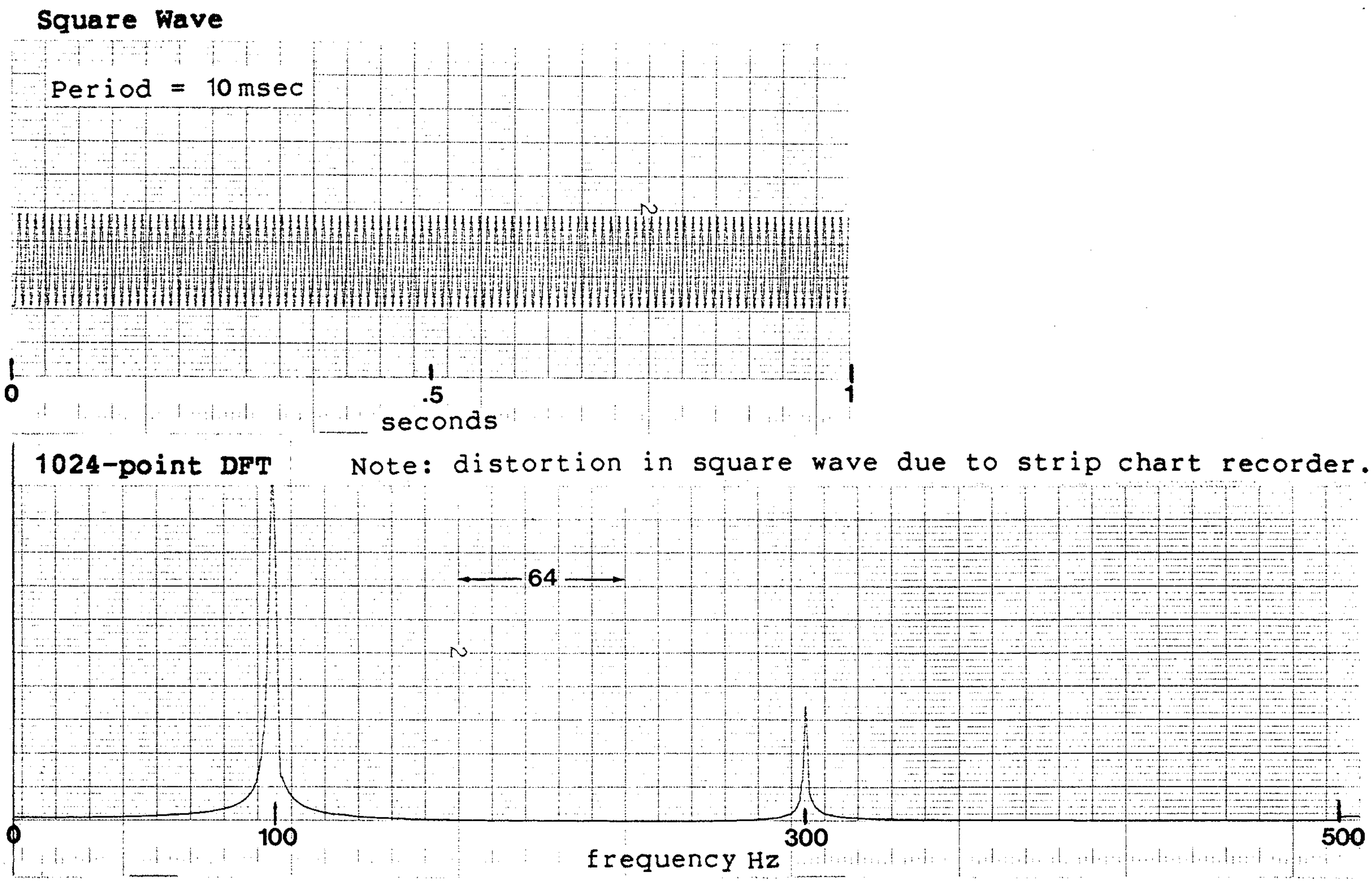


Figure 4.11: Square Wave and Corresponding 1024-point DFT

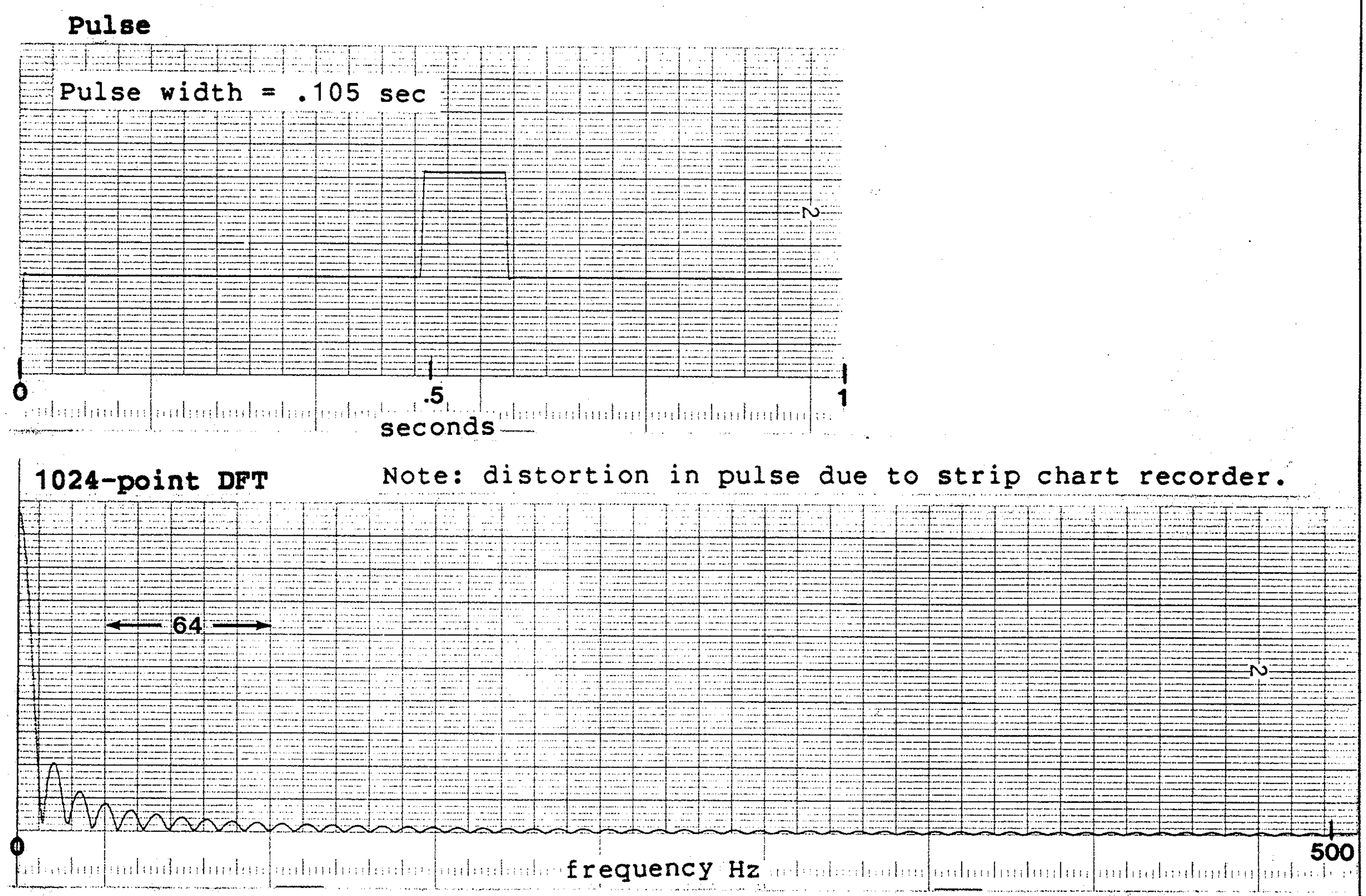


Figure 4.12: Pulse and Corresponding 1024-point DFT

4.6.4 PCG Energy Distribution Coefficient

To quantify the energy curve, a quantity which depends on the distribution of energy over the duration of the signal is used. This quantity is the Energy Distribution Coefficient [77] and is defined as

$$EDC_t = \sum_{n=1}^{L} a(n)E(n) / \sum_{n=1}^{L} E(n) \qquad (4.11)$$

where $E(n)$ is the energy curve, L is the number of samples in $E(n)$, $a(n)$ is a nondecreasing weighting sequence, and the subscript t denotes the time domain. In this work, we define $a(n) = n$. Additional subscripts s and d denote the systolic and diastolic segments, respectively. A similar quantity EDC_f is defined for the frequency domain with $E(n)$ in the above equation replaced by the energy spectrum. The denominator is a normalization factor.

From the above equation, it is apparent that due to the progressively heavier weighting PCG signals with systolic and/or diastolic murmurs will have larger EDC_{ts} and/or EDC_{td} values than a normal PCG signal with no murmurs. Also, signals with higher frequency components can be expected to have higher EDC_{fs} and EDC_{fd} values as their energy spectra will have larger values at frequencies away from the origin. Thus the PCG signal can be represented by the four quantities EDC_{ts}, EDC_{td}, EDC_{fs}, and EDC_{fd} which give an indication of the location of murmurs in the cardiac cycle and the frequency content of the signal. To compare different records of varying durations, a correction factor must be applied to the quantity EDC. In this work, the EDC_t values are corrected by dividing by the number of samples and multiplying by a factor of 1024. A correction factor is not necessary for EDC_f as the duration L of the energy spectrum is common to all signals, equal to one half the number of points used for the computation of the FFT.

4.7 SUMMARY

To summarize, an MC68000 microprocessor based system has been designed to analyze the time and frequency domain characteristics of the PCG signal. The ECG and carotid pulse signals are used as timing references for the PCG to identify the first and second heart sounds and to segment the PCG into systole and diastole. The systolic and diastolic energy curves and energy spectra are computed and quantified using the Energy Distribution Coefficient. Typical computation times for the PCG analysis system routines are shown in Table 4.2. These times could be considerably reduced with a faster microprocessor. Upon completion of the analysis routines the EDC values are available to the operator. A routine to display the results is then invoked allowing the operator to view the various output signals on an oscilloscope or strip chart recorder via the D/A converters.

TABLE 4.2

Typical Execution Times of the PCG Analysis System Routines

Signal Processing Routine	Typical Execution Time (seconds)
A/D Conversion	1.5
ECG Processing-QRS Complex Detection	1.8
Carotid Processing-Dicrotic Notch	0.6
PCG Processing-Time Domain:	
Systolic Energy Curve and EDC	2.2
Diastolic Energy Curve and EDC	3.5
PCG Processing-Frequency Domain:	
Systolic Energy Spectra and EDC	6.4
Diastolic Energy Spectra and EDC	6.4
TOTAL:	22.3

Chapter V

DISCUSSION OF RESULTS

Phonocardiogram signals of 5 healthy, normal subjects and 20 patients with valvular and other cardiovascular defects were taken up for study using the proposed techniques. For a majority of the patients, PCG signals were obtained from various recording sites on the chest wall. This provided a greater number of signals to be tested using the PCG analysis system. The ECG signal and carotid pulse were used as references for selection of the PCG signal over one cardiac cycle and for the segmentation of the PCG into systole and diastole. The energy curves and energy spectra of the systolic and diastolic segments of the PCG signals were obtained, and using these, EDC_{ts}, EDC_{td}, EDC_{fs}, and EDC_{fd} were computed.

An important aspect of PCG segmentation is to ensure minimal error results in the computation of the EDC values. Possible errors in the segmentation procedure include the following:

1. Portion of S2 included in systole.
2. Portion of systole included in diastole.
3. Portion of systole, with a systolic murmur, included in diastole.

The use of the standardized S2-D interval ensures that a portion of S2 is not included in the systolic segment, although the two other cases mentioned above are still possible sources which may affect the EDC values. Figures 5.1 and 5.2 show a normal PCG and a PCG containing a systolic murmur, respectively, with various magnitudes of PCG segmentation error. The PCG signals were initially viewed on a strip chart recording and the exact S2-D delay was used to segment the signal into systole and diastole. Segmentation errors were then introduced by varying the standardized S2-D interval. Referring to Table 4.1 it can be seen that the minimum S2-D delay is 35 msec. Using the standardized S2-D interval of 53 msec, the maximum error which could be expected in segmenting the PCG is 18 msec, i.e., 18 msec of systole will be included in the diastolic segment. This is one of the cases shown in Figures 5.1 and 5.2. The EDC_t and EDC_f values are computed for the case with no error and for the cases with segmentation errors. The results are reflected in Table 5.1. Note that the errors in the EDC values are relatively small, confirming the accuracy of this method. The PCG segmentation method proved to be quite accurate for the PCG analysis performed in this work.

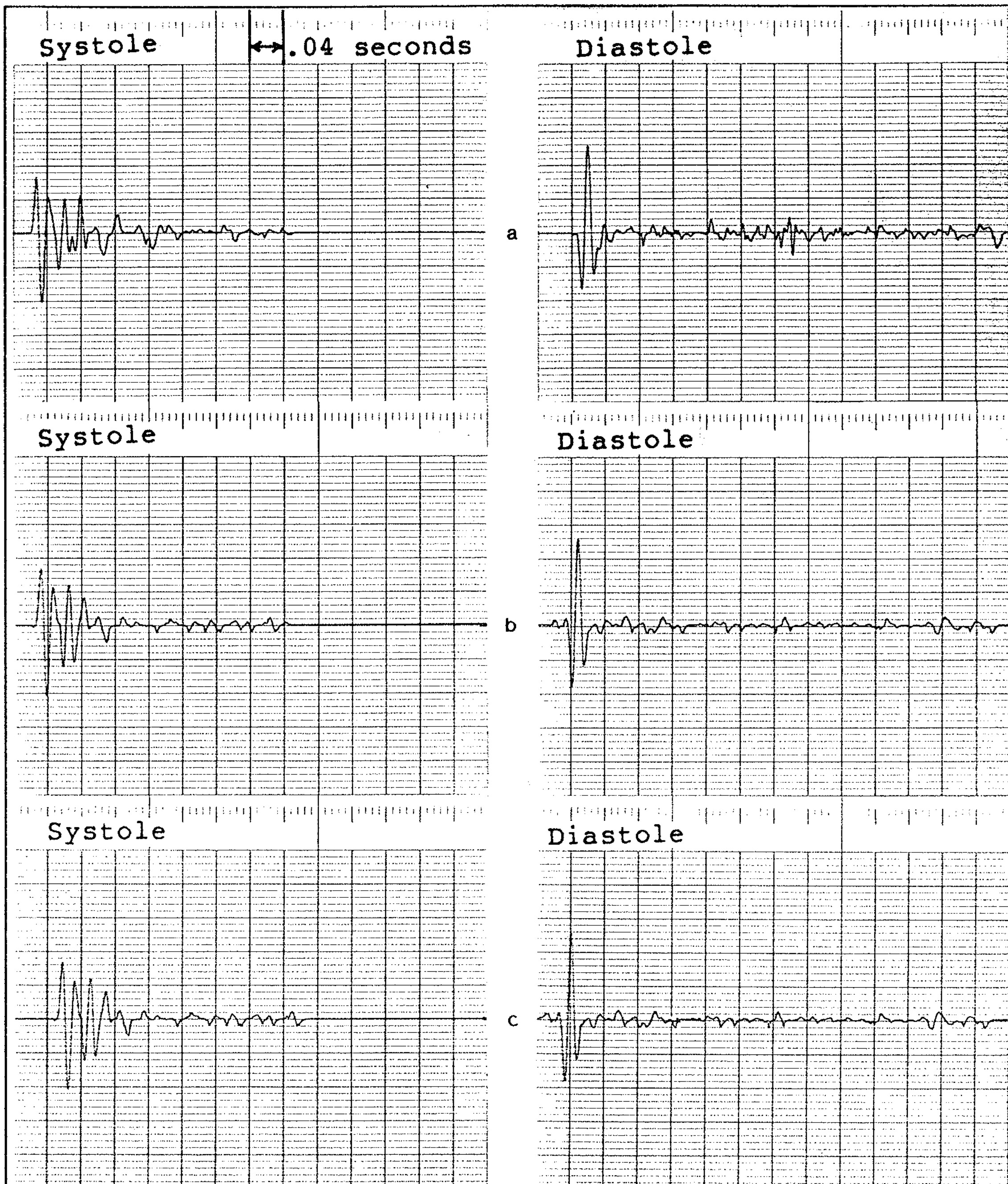


Figure 5.1: Normal PCG with Simulated Segmentation Errors

(a) ideal case:no portion of systole in diastolic segment;(b) error:10 msec of systole included in diastole (c) error:18 msec of systole included in diastole.

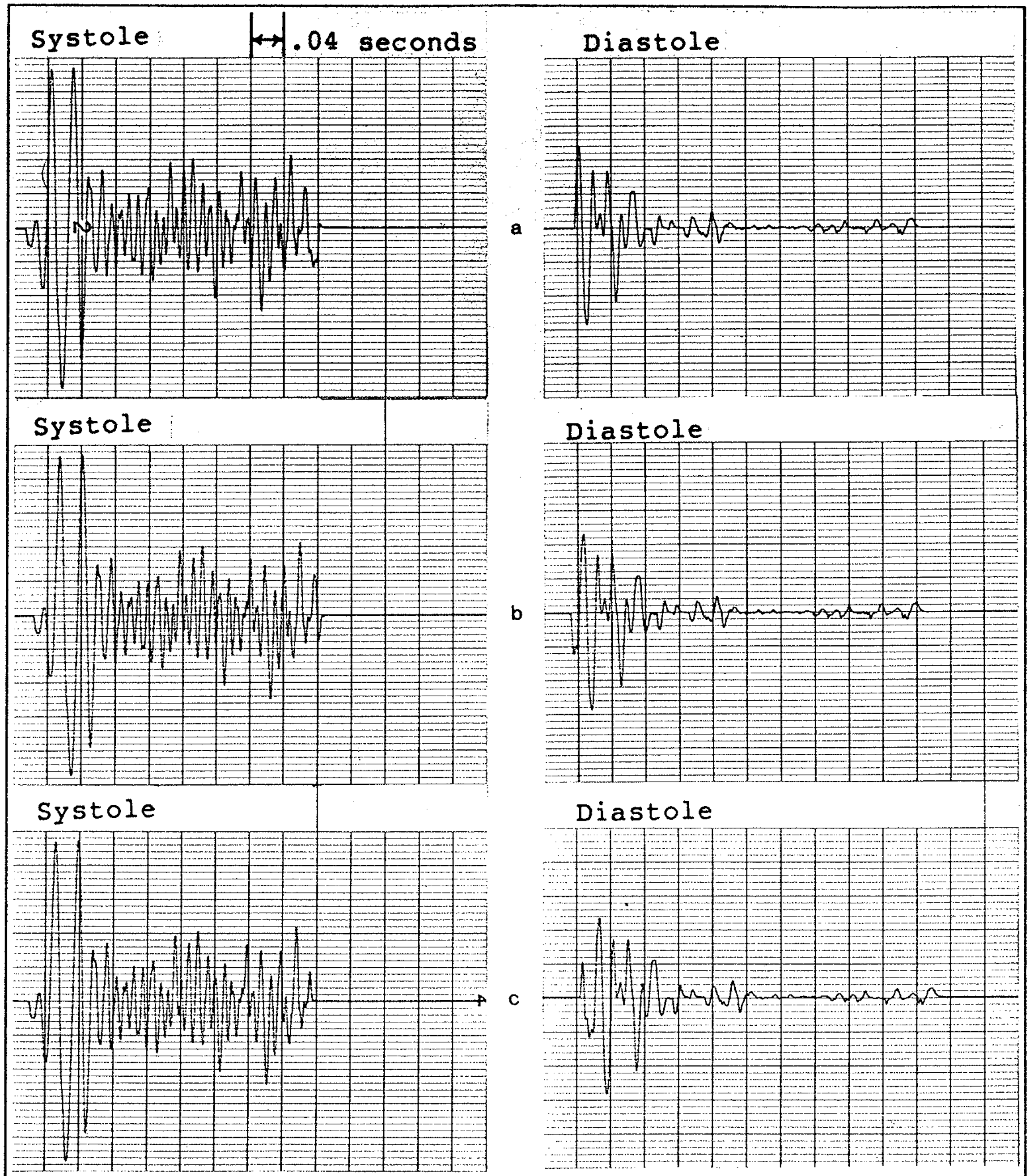


Figure 5.2: PCG with Systolic Murmur and Simulated Segmentation Errors

(a) ideal case:no portion of systole in diastolic segment;(b) error:10 msec of systole included in diastole (c) error:18 msec of systole included in diastole.

TABLE 5.1

EDC Errors Incurred From Segmentation Errors

PCG	Error	EDC_{ts}	EDC_{td}	EDC_{fs}	EDC_{fd}
Normal (Figure 5.1)	no error	160	123	77	82
	10 msec	163	128	77	80
	18 msec	170	132	77	80
SM (Figure 5.2)	no error	410	119	120	77
	10 msec	416	122	131	75
	18 msec	420	130	133	75

Figure 5.3 shows a PCG signal, segmented into systole and diastole, and the energy curve and energy spectrum of each segment. It can be seen that the energy curves display peaks corresponding to the first and second heart sounds. From the energy spectra it is seen that there is practically no energy beyond 120 Hz. The energy spectrum of the systolic segment contains peaks at approximately 25 Hz, 42 Hz, 58 Hz, and 74 Hz. It is interesting to note that a previous study for spectral decomposition of the first heart sound revealed it consists of a number of peaks in the frequency range 10 - 140 Hz [44]. In the present analysis, the peaks have appeared in the same frequency range. Similarly, the diastolic energy spectrum contains peaks at approximately 35 Hz, 54 Hz, 83 Hz, and 106 Hz, which agree with results attained in [45], in which spectral decomposition of the second heart sound was performed.

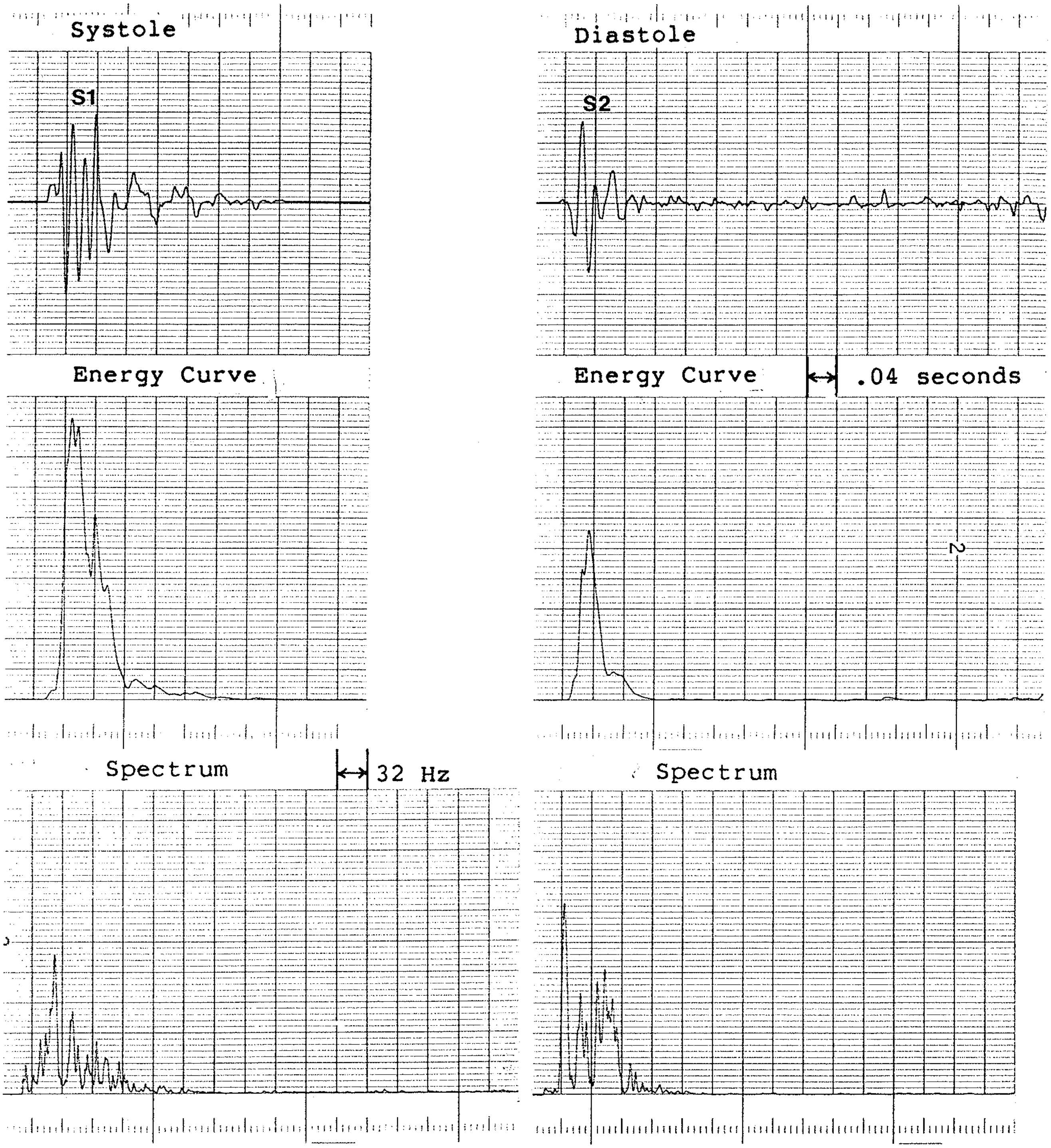


Figure 5.3: Normal PCG and Corresponding Energy Curves and Energy Spectra

The segmented PCG of a subject with ventricular septal defect (VSD) is shown in Figure 5.4. The corresponding energy curves display peaks corresponding to the first heart sound and a split second heart sound. The systolic energy curve shows the presence of a systolic murmur. The systolic energy spectrum indicates significant energy up to approximately 330 Hz, indicating the presence of a high frequency murmur. The diastolic energy curve and energy spectrum show a normal diastole with no murmurs and no significant energy beyond 120 Hz.

PCG signals of subjects with pansystolic ejection murmur (PEM), mitral insufficiency (MI), and aortic stenosis (AS) with corresponding energy curves and energy spectra are shown in Figures 5.5, 5.6, and 5.7. The systolic energy curve and energy spectrum shown in Figure 5.5 indicate the presence of a systolic murmur with energy up to approximately 200 Hz. The diastolic energy curve and energy spectrum indicate normal diastole. The systolic energy curve and the diastolic energy curve shown in Figure 5.6 display peaks corresponding to a systolic and diastolic murmur. However, these murmurs are of low frequency (below 120 Hz) and thus the energy spectra do not indicate significant energy at higher frequencies. The systolic and diastolic energy curves and energy spectra of the patient with AS, shown in Figure 5.7, exhibit the presence of a low frequency systolic murmur and a normal diastole.

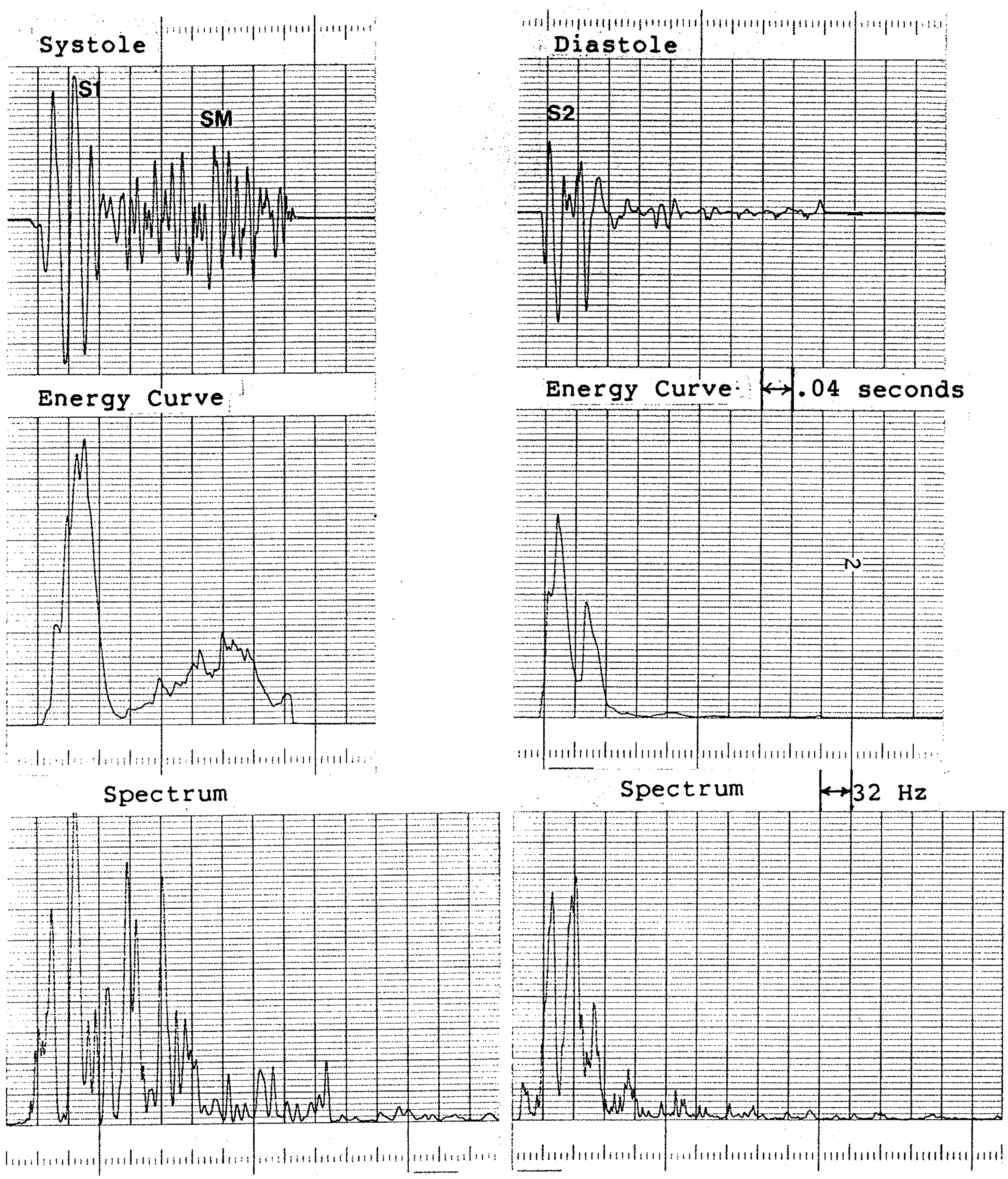


Figure 5.4: PCG with VSD and Corresponding Energy Curves and Energy Spectra

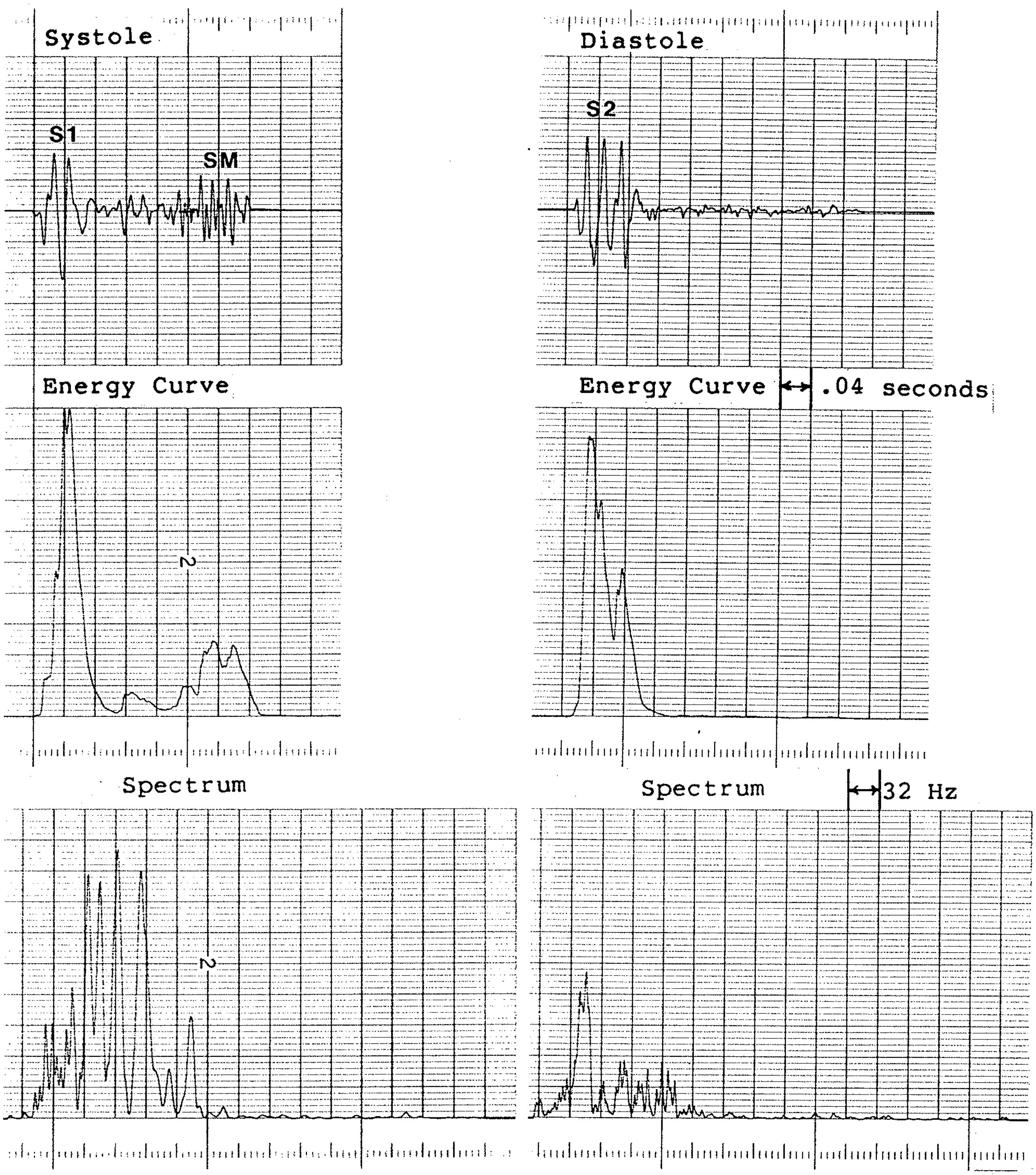


Figure 5.5: PCG with PEM and Corresponding Energy Curves and Energy Spectra

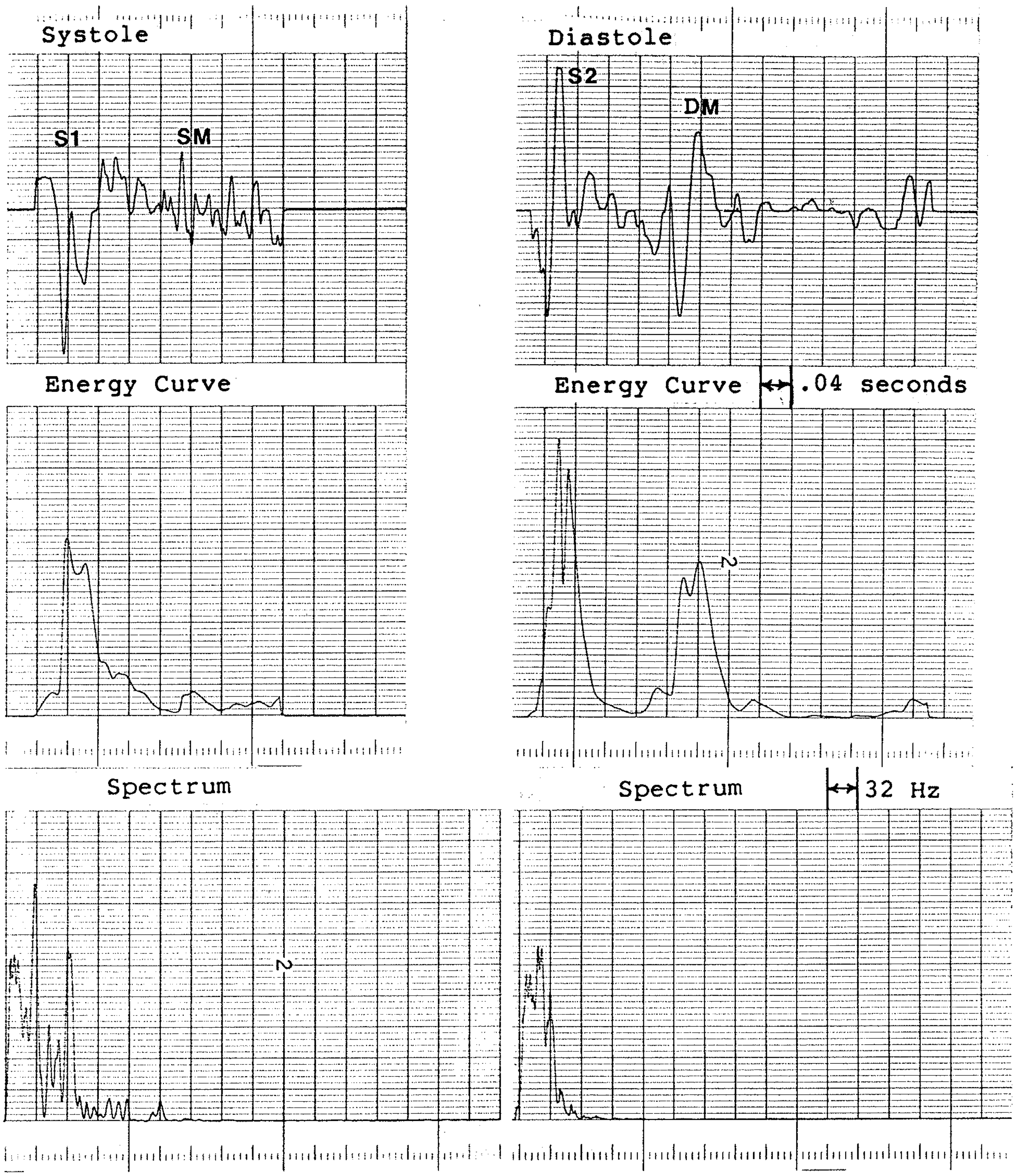


Figure 5.6: PCG with MI and Corresponding Energy Curves and Energy Spectra

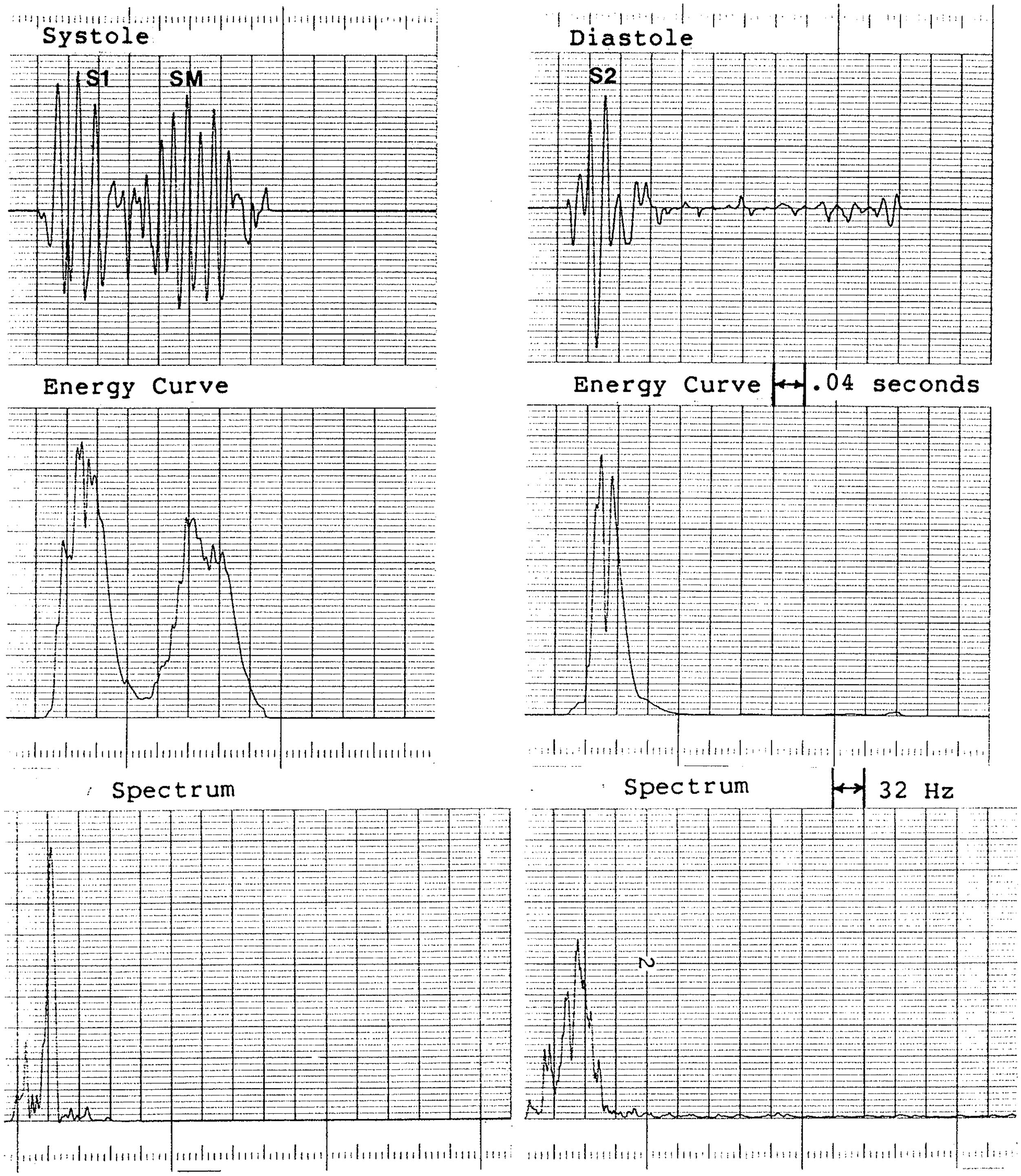


Figure 5.7: PCG with AS and Corresponding Energy Curves and Energy Spectra

To take into account variations of the PCG signal during inspiration and expiration, the average of a minimum of five sets of EDC_{ts}, EDC_{td}, EDC_{fs}, and EDC_{fd} values, obtained from PCG cycles picked up at random from the recordings, were computed for each subject. The average EDC_t and EDC_f values obtained for the 47 signals tested are shown in Table 5.2. The diagnosis of the cardiologist (col.2), the recording site of the PCG (col.3), and the observed features of the signal (col.4) are included in the table. From this table it can be seen that in general, signals with systolic murmurs and/or diastolic murmurs have greater EDC_{ts} and/or EDC_{td} values than for normal cases. Also the EDC_f values for the cases with high frequency murmurs are greater than those for normal cases. A definite classification scheme could not be developed due to the nonavailability of an adequate number of cases belonging to any particular disease category. However, the signals were divided into systole and diastole and further grouped into normal signals and signals with murmurs. The mean and standard deviation of the EDC_t and EDC_f values of each group are computed and reflected in Table 5.3. Here it is seen that the signals with systolic murmurs or diastolic murmurs have largers EDC_{ts} or EDC_{td} values, respectively, than those for normal signals. Murmurs are often graded depending upon their intensity. However, in this work the murmurs were not graded and murmurs of varying intensity are present within a group. This explains the considerably high standard deviations (particularly in the cases with murmurs). Other factors which could induce variations in the EDC_{td} values include a large or split second heart sound and the presence of a third or fourth heart sound in late diastole.

For systole it can be seen that the EDC_{fs} values are greater for cases with systolic murmurs than for normal cases. However, many of the signals of the pathological cases did not have apparent energy beyond 120 Hz. A few signals had energy extending up to 330 Hz, these resulting in large EDC_{fs} values and thereby increasing the EDC_{fs} mean. For the EDC_{fd} values none of the pathological cases had diastolic murmurs with significant energy over 120 Hz; i.e. all cases had low frequency diastolic murmurs.

With an adequate training set having a sufficiently large number of cases in each category and gradation of murmurs, a pattern recognition technique could be developed with the EDC_t and EDC_f values as parameters for detection and classification of murmurs.

TABLE 5.2

Average EDC_t and EDC_f Values

Patient	Diagnosis	Site	Observed features	EDC_{ts}	EDC_{td}	EDC_{fs}	EDC_{fd}
1	2	3	4	5	6	7	8
A	N	P	N	198	149	61	67
B	N	P	N	169	154	75	80
C	N	P	N	181	151	50	60
1A	MI	P	SM,MDM	223	180	55	56
1B		T		207	172	55	56
2A	MI	P	SM,DM	379	154	52	54
2B		T	SPLS2	205	188	50	52
2C		M		258	273	50	52
3A	VSD	P	SM,SPLS2	540	120	104	100
3B		T	S3	520	188	108	96
4A	MR	P	SM	327	180	54	56
4B		T		205	190	56	54
5	ASD	P	SM,SPLS2	262	196	80	68
6	PAV	P	SM	259	138	70	65
7A	VSD	P	SM,SPLS2	469	172	116	70
7B		T		390	142	125	68
8A	AS	A	SM	340	148	80	74
8B		T		330	130	69	70
9A	VSD	P	SM	245	197	74	56
9B		T		204	156	65	60
10	VSD	T	SM	200	173	60	69
11A	PEM	P	SM	356	144	118	80
11B		C		403	200	95	66
12A	VSD,PS	P	SEM,RA	315	230	74	89
12B		P	RESP	350	230	80	62
12C		M		326	230	86	61
13	VSD	T	SM	412	196	103	55
14A	N	P	N	170	160	50	48
14B		A		175	145	54	50
15A	AS	A	SM	454	186	74	56
15B		M		381	181	74	56
15C		P		360	143	60	58
16A	VSD	P	PSM	430	176	109	75
16B		T		458	143	103	60
16C		T		421	159	105	65
17	ASD,PS	T	SM,DM	253	266	82	80
18	N	T	N	168	149	56	54
19A	VSD,PS	P	SM	490	237	112	70
19B		T		433	183	108	68
20	PS	A	MSM	230	144	68	75
21	PDA	P	SM,DM	346	318	58	69

1	2	3	4	5	6	7	8
22A	N	P	N	205	148	80	82
22B		T		187	166	76	80
22C		M		179	157	75	78
23A	TA,PS	A	SM,DM	416	364	76	80
24A	AR	P	SM,DM	224	155	84	76
24B		M		246	263	73	82

Legend: col.2: N-normal, AS-aortic stenosis

ASD-atrial septal defect

MI-mitral insufficiency, MR-mitral regurgitation

PAV-prosthetic aortic valve

PDA-patent ductus arteriosus

PEM-pulmonary ejection murmur

PS-pulmonary stenosis, TA-tricuspid artesia

VSD-ventricular septal defect

col.3: P-pulmonary area A-aortic area

T-tricuspid area M-mitral area

C-clavicular

col.4: N-normal, SM-systolic murmur

DM-diastolic murmur

MDM/SM-middiastolic/systolic murmur

PSM-pansystolic murmur

RESP-respiration pulse

SPLS2-split second heart sound

S3-third heart sound

TABLE 5.3

Mean and SD of EDC_t and EDC_f Values

Systole	Number of cases	EDC_{ts}	SD	EDC_{fs}	SD
Normal	9	181.3	12.4	64.1	11.6
SM	38	338.6	97.4	80.7	21.6
Diastole	**Number of cases**	**EDC_{td}**	**SD**	**EDC_{fd}**	**SD**
Normal	37	164.7	24.5	67.6	12.1
DM	10	233.2	69.8	65.7	12.2

Chapter VI

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY

6.1 CONCLUSIONS

In conclusion a prototype MC68000 microprocessor-based system has been designed to provide quantitative analysis of the phonocardiogram. The system includes 32K bytes RAM, A three-channel data acquisition section, an arithmetic processing unit, and a set of D/A converters. The system has the provisions to be dedicated to phonocardiographic applications. A detection algorithm for the first and second heart sounds, which is one of the most important problems in an automatic phonocardiogram diagnosis system, has been developed. The ECG and carotid pulse signals are used to segment the phonocardiogram into systole and diastole. A method has been presented to quantify the systolic and diastolic segments of the phonocardiogram into four parameters representing their time and frequency domain characteristics. Tests on 47 phonocardiogram signals show that the method can lead to an interesting and potentially useful method for detection and classification of murmurs. Studies with a large number of signals could lead to an elaborate classification scheme. A system performing such classification would be extremely useful for screening purposes and routine diagnosis.

6.2 RECOMMENDATIONS

1. A large number of signals in each disease category should be tested using the PCG analysis system and the methods presented in this work. To improve classification the murmurs should be graded according to their intensity. Furthermore, when recording the PCG signals, the transducer should be placed on the area of the chest at which the intensity of the murmur is maximized. The signals should be recorded in an acoustically quiet room, if possible. A pattern classification technique could then be developed using the EDC values as parameters for the detection and classification of murmurs.

2. The PCG analysis system could be used to investigate other areas of heart sound research such as:

   a) applying passive sonar techniques to PCG signals recorded at multiple locations on the thorax to estimate the location of heart sounds in 3D. This work would involve identification of corresponding events in the different signals by cross-correlation and coherence techniques; estimation of time delay between arrival of the events at different locations; and, ranging to determine the locations of sources.

b) testing the functional and structural integrity of cardiac prosthetic valves using the sounds produced by the valves.

3. Modifications to the hardware could be undertaken to increase the efficiency of the PCG analysis system. In particular, a direct memory access controller could be added to increase the speed of APU operations. In addition, the system RAM could be increased in order to store longer durations of signal. The added RAM would also provide the additional memory required to allow the signals to be digitized at a higher sampling rate.

REFERENCES

[1] H. Garret deYoung, "State of the heart", High Technology, Vol.4, pp.33-41, May,1984.

[2] A. Macovski, Medical Imaging Systems. Englewood Cliffs, New Jersey: Prentice Hall, 1983.

[3] R.F. Rushmer, Cardiovascular Dynamics, 4th Ed. Philadelphia: W.B. Saunders Company, 1976.

[4] J.K. Lewis and W. Dock, "The origin of heart sounds and their variations in myocardial disease", Journal of the American Medical Association, Vol.110, pp.271-275, January,1938.

[5] H.A. Sacks and R.C. Roberts, "Heart sounds in young adults", American Heart Journal, Vol.18, pp.303-307, September,1939.

[6] A.A. Luisada, C.K. Liu, C. Aravanis, M. Testelli, and J. Morris, "On the mechanism of production of heart sounds", American Heart Journal, Vol.55, pp.383-399, March,1958.

[7] D.J. Coleman, "On the origin of the second heart sound", American Heart Journal, Vol.65, pp.237-239, February,1963.

[8] G. Di Bartolo, D. Nunez-Dey, G. Muiesan, D.M. MacCanon, and A.A. Luisada, "Hemodynamic correlates of the first heart sound", American Journal of Physiology, Vol.201, pp.888-892, November,1961.

[9] P.M. Shah, M. Mori, D.M. MacCanon, and A.A. Luisada, "Hemodynamic correlates of the various components of the first heart sound", Circulation Research, Vol.12, pp.386-392, April,1963.

[10] T. Sakamoto, R. Kusukawa, D.M. MacCanon, and A.A. Luisada, "Hemodynamic determinants of the amplitude of the first heart sound", Circulation Research, Vol.16, pp.45-57, January,1965.

[11] T.E. Piemme, G.O. Barnett, and L. Dexter, "Relationship of heart sounds to acceleration of blood flow", Circulation Research, Vol.18, pp.303-315, March,1966.

[12] A.A. Luisada, D.M. MacCanon, B. Coleman, and L.P. Feigen, "New studies on the first heart sound", American Journal of Cardiology, Vol.28, pp.140-149, August,1971.

[13] A.A. Luisada, "The second heart sound in normal and abnormal conditions", American Journal of Cardiology, Vol.28, pp.150-161, August,1971.

[14] A.A. Luisada and D.M. MacCanon, "The phases of the cardiac cycle", American Heart Journal, Vol.83, pp.705-711, May,1972.

[15] A.A. Luisada, D.M. MacCanon, S. Kumar, and L.P. Feigen, "Changing views on the mechanism of the first and second heart sounds", American Heart Journal, Vol.88, pp.503-514, October,1974.

[16] H.N. Sabbah and P.D. Stein, "Investigation of the theory and mechanism of the origin of the second heart sound", Circulation Research, Vol. 39, pp.874-882, December,1976.

[17] P.D. Stein and H.N. Sabbah, "Origin of the second heart sound: clinical relevance of new observations", American Journal of Cardiology, Vol.41, pp.108-110, January,1978.

[18] K.O. Lim, Y-C Liew, and C-H Oh, "Analysis of mitral and aortic valve vibrations and their role in the production of the first and second heart sounds", Physics in Medicine and Biology, Vol.25, pp.727-733, July,1980.

[19] A.A. Luisada and F. Portaluppi, The Heart Sounds: New Facts and Their Clinical Applications. New York: Praeger Publishers, 1982.

[20] L.-G. Durand and R. Guardo, "A computer model for the study of left ventricular wall contributions to the first heart sound", pp. 445-448 in Computers in Cardiology, IEEE, New York, 1982.

[21] L.P. Feigen, "Physical characteristics of sound and hearing", American Journal of Cardiology, Vol.28, pp.130-133, August,1971.

[22] J.S. Butterworth and E.H. Reppert, "Auscultatory acumen in the general medical population", Journal of the American Medical Association, Vol.174, pp.32-34, September,1960.

[23] R.J. Dobrow, J.B. Calatayud, and S. Abraham, "A study of physician variation in heart sound interpretation", Medical Annals District of Columbia, Vol.33, pp.305-308, July,1964.

[24] J.F. Green, Fundamental Cardiovascular and Pulmonary Physiology: An Integrated Approach for Medicine. Philadelphia: Lea and Febiger, 1982.

[25] M.E. Tavel, Clinical Phonocardiography and External Pulse Recording, 3rd Ed. Chicago: Year Book Medical Publishers Inc., 1978.

[26] E. Van Vollenhoven, N. Suzumura, D.N. Ghista, J. Mazumdar, and T. Hearn, "Phonocardiography: Analyses of instrumentation, and vibration of heart structures to determine their constitutive properties", pp.68-118 in Advances in Cardiovascular Physics, Vol.2, Karger, Basel, Switzerland, 1979.

[27] Application Note AN732, "Phonocardiography and external pulse recording", Hewlett Packard, 1973.

[28] J.G. Webster, Editor, Medical Instrumentation: Application and Design. Boston: Houghton Mufflin Company, 1978.

[29] L. Karpman, J. Cage, C. Hill, A.D. Forbes, V. Karpman, and K. Cohn, "Sound envelope averaging and the differential diagnosis of systolic murmurs", American Heart Journal, Vol.90, pp.600-606, November,1975.

[30] A.A. Sarkady, R.R. Clark, and R. Williams, "Computer analysis techniques for phonocardiogram diagnosis", Computers and Biomedical Research, Vol.9, pp.349-363, August,1976.

[31] D.S. Gerbarg, F.W. Holcomb, J.J. Hofler, C.E. Bading, G.L. Schultz, and R.E. Sears, "Analysis of phonocardiogram by a digital computer", Circulation Research, Vol.11, pp.569-576, September,1962.

[32] M. Yokoi, Z. Uozumi, N. Okamoto, Y. Mizuno, T. Iwatsuka, H. Takahashi, Y. Watanabe, and S. Yashui, "Clinical evaluation on 5 years' experience of automated PCG analysis", Japan Heart Journal, Vol.18, pp.482-490, July,1977.

[33] E. Van Vollenhoven, A. Van Rotterdam, T. Dorenbos, and F.G. Schlesinger, "Frequency analysis of heart murmurs", Medical and Biological Engineering, Vol.7, pp.227-233, March,1969.

[34] D.S. Gerbarg, "Computer analysis of phonocardiogram", Progress in Cardiovascular Disease, Vol.5, pp.393-405, January,1963.

[35] N.S. Blackman, S. Blumenthal, K.D. Brownell, J. Wolfson, and D. Harris, "Cardiac screening by computerized auscultation", American Journal on Public Health, Vol.59, pp.1177-1187, July,1969.

[36] W. Litwon and F. Begon, "Digital processing of phonocardiograms: first results", International Journal of Biomedical Engineering, Vol.5, pp.58-69, January,1974.

[37] V.A. McKusick, S.A. Talbot, and G.N. Webb, "Spectral phonocardiography", "Spectral phonocardiography", Bulletin of the Johns Hopkins Hospital, Vol.94, p.187-198, 1954.

[38] D.E. Winer, L.W. Perry, and C.A. Caceres, "Heart sound analysis: A three dimensional approach", American Journal of Cardiology, Vol.16, pp.547-551, October,1965.

[39] R.J. Adolph, J.F. Stephens, and K. Tanaka, "The clinical value of frequency analysis of the first heart sound in myocardial infarction", Circulation, Vol.41, pp.1003-1014, June,1970.

[40] A. Sakai, L.P. Feigen, and A.A. Luisada, "Frequency distribution of the heart sounds in normal man", Cardiovascular Research, Vol.5, pp.358-363, July,1971.

[41] W.B. Clarke, S.M. Austin, P.M. Shah, P.M. Griffen, J.T. Dove, J. McCullough, and B.F. Schreiner, "Spectral energy of the first heart sound in acute myocardial ischemia", Circulation, Vol.57, pp.593-598, March,1978.

[42] E.L. Frome and E.L. Frederickson, "Digital spectrum analysis of the first and second heart sounds", Computers and Biomedical Research, Vol.7, pp.421-431, October,1974.

[43] A.P. Yoganathan, R. Gupta, and W.H. Corcoran, "Fast Fourier transform in the analysis of biomedical data", Medical and Biological Engineering, Vol.14, pp.239-244, March,1976.

[44] A.P. Yoganathan, R. Gupta, F.E. Udwadia, J.W. Miller, W.H. Corcoran, R. Sarma, J.L. Johnson, and R.J. Bing, "Use of the fast Fourier transform for frequency analysis of the first heart sound in normal man", Medical and Biological Engineering, Vol.14, pp.69-73, January,1976.

[45] A.P. Yoganathan, R. Gupta, F.E. Udwadia, W.H. Corcoran, R. Sarma, and R.J. Bing, "Use of the fast Fourier transform in the frequency analysis of the second heart sound in normal man", Medical and Biological Engineering, Vol.14, pp.455-459, July,1976.

[46] K.J. Blinowska, M.A. Czerminska, O.A. Chomicki, and T. Gorowski, "Computer analysis of the frequency spectrum of a phonothyreogram", Medical and Biological Engineering and Computing, Vol.17, pp.207-210, March,1979.

[47] P. Pinna Pintor, E. Piccolo, S. Bartolozzi, and V. Fontana, "The FFT in the study of the fourth heart sound", pp.369-372 in Computers in Cardiology, Sept. 26-28, 1979, Geneva, IEEE, New York, 1979.

[48] C. Longhini, F. Portaluppi, E. Arslen, F. Pedrelli, "The FFT in the analysis of the normal phonocardiogram", Japan Heart Journal, Vol.20, pp.333-339, 1979.

[49] D. Nandagopal and J. Mazumdar, "A study of temporal variation in heart sound frequency spectra using fast Fourier transform", Australasian Physical and Engineering Sciences in Medicine, Vol.4, pp.47-50, 1981.

[50] T.C. Hearn, J. Mazumdar, and L.J. Mahar, "First heart sound spectra in relation to anterior mitral-leaflet closing velocity", Medical and Biological Engineering and Computing, Vol.20, pp.466-472, July,1982.

[51] H. Weed, M. Sauter, R. Roberts, G. Vossius, and H. Kwee, "Relationship of acoustic heart sound frequency shift to heart function", pp. 386-389 in IEEE 1981 Frontiers in Engineering in Health Care, ed. B.A. Cohen, IEEE, New York, 1981.

[52] D. Nandagopal, J. Mazumdar, G. Karolyi, and T. Hearn, "An instrumentation system and analysis procedure for phonocardiographic studies", Australian Journal of Biomedical Engineering, Vol.2, pp.16-20, March,1981.

[53] A. Iwata, N. Suzumura, and K. Ikegaya, "Pattern classification of the phonocardiogram using linear prediction analysis", Medical and Biological Engineering and Computing, Vol.15, pp.407-412, July,1977.

[54] A. Iwata, N. Ishii, N. Suzumura, and K. Ikegaya, "Algorithm for detecting the first and second heart sounds by spectral tracking", Medical and Biological Engineering and Computing, Vol.18, pp.19-26, January,1980.

[55] D. Nandagopal, J. Mazumdar, R.E. Bogner, and E. Goldblatt, "Spectral analysis of the second heart sound in normal children by selective linear prediction coding", Medical and Biological Engineering and Computing, Vol.22, pp.229-239, May,1984.

[56] M. Okada, "Chest wall maps of heart sounds and murmurs", Computers and Biomedical Research, Vol.15, pp.281-294, June,1982.

[57] R. Beyar, S. Braun, S. Levkovitch, and Y. Palti, "A new statistical approach to heart-sound processing", pp.376-380 in IEEE Frontiers of Engineering and Computing in Health Care, Columbus, Ohio, 1983.

[58] R. Beyar, S. Levkovitz, S. Braun, and Y. Palti, "Heart-sound processing by average and variance calculation - physiologic basic and clinical implications", IEEE Transactions on Biomedical Engineering, Vol. BME-31, pp.591-596, September, 1984.

[59] D.W. Suobank, A.P. Yoganathan, E.C. Harrison, and W.H. Corcoran, "A quantitative method for the in vitro study of sounds produced by prosthetic aortic heart valves I: analytical considerations." Medical and Biological Engineering and Computing, Vol.22, pp.32-39, January,1984.

[60] D.W. Suobank, A.P. Yoganathan, E.C. Harrison, W.H. Corcoran, "A quantitative method for the in vitro study of sounds produced by prosthetic aortic heart valves II: An experimental, comparative study of the sounds produced by a normal and simulated-abnormal Starr-Edwards series 2400 aortic prosthesis", Medical and Biological Engineering and Computing, Vol.22, pp.40-47, January,1984.

[61] D.W. Suobank, A.P. Yoganathan, E.C. Harrison, W.H. Corcoran, "A quantitative method for the in vitro study of sounds produced by prosthetic aortic heart valves III: An experimental, comparative study of the sounds produced by normal and simulated-abnormal Smeloff aortic prostheses", Medical and Biological Engineering and Computing, Vol.22, pp.48-54, January,1984.

[62] J.C. Hylen, F.E. Kloster, R.H. Herr, P.Q. Hull, A.W. Ames, A. Starr, and H.E. Griswold, "Phonocardiographic diagnosis of aortic ball variance", Circulation, Vol.38, pp.90-102, July,1968.

[63] J.T. Willerson, J.A. Kastor, R.E. Dinsmore, E. Mundth, and M.J. Buckley, W.G. Austen, and C.A. Sanders, "Non-invasive assessment of prosthetic mitral paravalvular and intravalvular regurgitation", British Heart Journal, Vol.34, pp.561-568, June,1972.

[64] E.N. Mercer, B.D. McCallister, E.R. Giuliani, R.S. Zitnick, "Aortic opening to closing ratio on phonocardiograms of patients with Starr-Edwards aortic valve prosthesis", Mayo Clinic Proceedings, Vol.47, pp.42-47, January,1972.

[65] O.W. Boicurt, J.D. Bristow, A. Starr, H.E. Griswold, "A phonocardiographic study of patients with multiple Starr-Edwards prosthetic valves", British Heart Journal, Vol.28, pp.531-538, April,1966.

[66] R.J. Romanoff, H.W. Seipp,Jr., L.C. Amsler, E.H. Johnson, and C.L. McIntosh, "Computer-assisted detection of prosthetic aortic heart valve dysfunction", Computers and Biomedical Research, Vol.10, pp.35-44. February,1977.

[67] P.D. Stein, H.N. Sabbah, J.B. Lakier, and S. Goldstein, "Frequency spectrum of the aortic component of the second heart sound in patients with normal valves, aortic stenosis and aortic porcine xenografts", American Journal of Cardiology, Vol.46, pp.48-52, July,1980.

[68] P.D. Stein, H.N. Sabbah, J.B. Lakier, D.J. Magilligan,Jr., and S. Goldstein, "Frequency of the first heart sound in the assessment of stiffening of mitral bioprosthetic valves", Circulation, Vol.63, pp.200-203, January,1981.

[69] T.H. Joo, J.H. McClellan, R.A. Foale, G.S. Myers, and R.S. Lees, "Pole-zero modeling and classification of phonocardiograms", IEEE Transactions on Biomedical Engineering, Vol.BME-30, pp.110-117, February,1983.

[70] L.-G. Durand, J. de Guise, and R. Guardo, "FFT techniques for the spectral analysis of prosthetic heart valve sounds", pp.1105-1111 in Proceedings of the 4th Annual Symposium on Computer Applications in Medical Care, IEEE, Washington, D.C., 1980.

[71] L.G. Durand, M. Brais, M. Blanchard, M. Guerin, M Sleger, J. de Guise, and R. Guardo, "Fourier analysis of Ionescu-Shiley prosthetic closing sounds", p. 150 in Proceedings of the 37th Annual Conference on Engineering in Medicine and Biology, Los Angeles, California, 1984.

[72] M.R. Rangaraj and I.S.N. Murthy, "Analysis and modelling of phonocardiogram", pp.8-53 to 8-56 in Proceedings of the 1st Mediterranean Conference on Medical and Biological Engineering, Sorrento, 1977.

[73] M.R. Rangaraj and I.S.N. Murthy, "Cepstral filtering of phonocardiogram signal", pp.118-119 in Digest of the International Symposium and Workshop on Biomedical Engineering, Indian Institute of Technology, New Delhi, 1978.

[74] M.R. Rangaraj, Digital Analysis of the ECG, PCG, and EMG Signals, Department of Electrical Engineering, Indian Institute of Science, Ph.D. Thesis, Bangalore, 1979.

[75] M.R. Rangaraj and I.S.N. Murthy, "New techniques for phonocardiogram analysis", Medical and Life Sciences Engineering, Vol.5, pp.6-17, 1979.

[76] M.R. Rangaraj and I.S.N. Murthy, "Classification of PCG via pole-zero modelling", pp.S7-26 to S7-29 in Proceedings of the 7th All India Symposium on Biomedical Engineering, Hyderabad, 1978.

[77] M.R. Rangaraj and I.S.N. Murthy, "Quantitative analysis of the phonocardiogram for detection of murmurs", Journal of Biomedical Engineering, Vol.1, pp.247-252, 1979.

[78] R.J. Lehner, A Microprocessor Based Heart Sound Analyzer, Department of Electrical Engineering, University of Manitoba, B.Sc. Thesis, Winnipeg, 1983.

[79] R.J. Lehner and R.M Rangayyan, "A microcomputer system for quantitative analysis of the phonocardiogram", in Proceedings MEDCOMP-83, IEEE, Athens, Ohio, 1983.

[80] M.R. Rangaraj and D.N. Dutt, "Speech processing techniques for the analysis of heart sounds", pp.S4-9 to S4-12 in Proceedings of the 7th All India Symposium on Biomedical Engineering, Hyderabad, 1978.

[81] H.V. Pippberger, R.A. Dunn, A.S. Berson, "Computer methods in electrocardiography", Annual Review of Biophysics and Bioengineering, Vol.4, pp.15-42, 1975.

[82] J.M. Jenkins, "Computerized electrocardiography", CRC Critical Reviews on Bioengineering, pp.307-350, November,1981.

[83] M. Okada, "3-dimensional graphic display of elctrocardiograms", Japan Circulation Journal, Vol.40, pp.167-175, 1976.

[84] L.J. Thomas, K.W. Clark, C.N. Mead, K.L. Ripley, B.F. Spenner, and G.C. Oliver, "Automated cardiac dysrhythmia analysis", Proceedings of the IEEE, Vol.67, pp.1322-1337, September,1979.

[85] Y. Yamashita, "Theoretical studies on the inverse problem in electrocardiography and the uniqueness of the solution", IEEE Transaction on Biomedical Engineering, Vol.BME-29, pp.719-725, November,1982.

[86] K. Yajima, S. Kinoshita, H. Tanaka, T. Ihara, and T. Furukawa, "Body surface potential mapping system equipped with a microprocessor for the dynamic observation of potential patterns", Medical and Biological Engineering and Computing, Vol.21, pp.83-90, January,1983.

[87] J.R. Cox,Jr., F.M. Nolle, and R.M. Arthur, "Digital analysis of the electoencephalogram, the blood pressure, and the electrocardiogram", Proceedings of the IEEE, Vol.60, pp.1137-1164, October,1972.

[88] I.S.N. Murthy and M.R. Rangaraj, "New concepts for PVC detection", IEEE Transactions on Biomedical Engineering, Vol.BME-26, pp.409-415, July,1979.

[89] P.H. Pan and A.W. Bennet, "Improved transform techniques for electrocardiogram analysis", p.92 in 33rd Annual Conference on Engineering in Medicine and Biology, Washington, D.C., 1980.

[90] H.A. Fozzard, P. Kinias, and A.L. Pai, "Algorithms for analysis of online pressure signals", in Computers in Cardiology, IEEE, pp.77-80, 1974.

[91] T.A. Pryor, "Some techniques for extraction of useful features from biological signals", pp.1-3 in Proceedings of the 5th Hawaii International Conference on Systems, Sciences - Computers in Biomedicine, 1972.

[92] J.R. Cox, F.M. Nolle, H.A. Fozzard, and G.G. Oliver,Jr., "AZTEC, a preprocessing program for real-time ECG rhythm analysis", IEEE Transactions on Biomedical Engineering, Vol.BME-15, pp.128-129, April,1968.

[93] M.E. DeBakey and D.H. Glaeser, "Waveform analysis of the central arterial pressure using a technique of data compression", Journal of the Association for the Advancement of Medical Instrumentation, Vol.6, pp.60-64, January,1972.

[94] M.C. Kyle, J.D. Klingeman, and E.D. Freis, "Computer identification of brachial arterial pulse waves", Computers and Biomedical Research, Vol.2, pp.151-159, October,1968.

[95] E.D. Freis and M.C. Kyle, "Computer analysis of carotid and brachial pulse waves: effects of age in normal subjects", American Journal of Cardiology, Vol.22, pp.691-695, November,1968.

[96] G. Stockman, L. Kanal, and M.C. Kyle, "Structural pattern recognition of carotid pulse waves using a general waveform parsing system", Communications of the Association of Computing Machines, Vol.19, pp.688-695, December,1976.

[97] P. Kinias, H.A. Fozzard, and M.J. Norusis, "A real-time pressure algorithm", Computers and Biomedical Research, Vol.11, pp.211-220, April,1981.

[98] C.F. Starmer, P.A. McHale, and J.C. Greenfield, Jr., "Processing of arterial pressure waves with a digital computer", Computers and Biomedical Research, Vol.6, pp.90-96, February,1973.

[99] K. Skrabie, Ed., MC68000 16-bit Microprocessor User's Manual, 4th Ed. Englewood Cliffs, New Jersey: Prentice-Hall, 1982.

[100] MC68000 Educational Computer Board User's Manual, 2nd Ed. Tempe, Arizona: Motorola, 1982.

[101] MC68230 Parallel Interface/Timer. Austin, Texas: Motorola, 1975.

[102] MC68000 Cross Assembler Reference Manual, 3rd Ed. Tempe, Arizona: Motorola, 1983.

[103] R.J. Lehner, Use of the Carotid Pulse for Identification of the Second Heart Sound. Digital Signal Processing project, Department of Electrical Engineering, University of Manitoba, 1984.

[104] R.J. Lehner and R.M. Rangayyan, "Use of the carotid pulse for detection of the second heart sound", in Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India, December,1984.

[105] M.B. Rappaport and H.B. Sprague, "Physiologic and physical laws that govern auscultation and their clinical application", American Heart Journal, Vol.21, pp.257-318, March,1941.

[106] R.W. Schafer and L.R. Rabiner, "Digital representations of speech signals", Proceedings of IEEE, Vol.63, pp.662-677, April,1975.

[107] A.W. Oppenheim and R.W. Schafer, Digital Signal Processing. Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1975.

[108] L.G. Durand (Clinical Research Institute of Montreal, Quebec), Discrete Fourier transform Subroutine. private communication.

[109] F.J. Taylor, Digital Filter Design Handbook. New York: Marcel Dekker, Inc, 1983.

Appendix A

SAAB SCHEMATIC DIAGRAMS

TABLE A.1

Device Reference

Reference	Type	GND	+5V	-5V	+12V	-12V
U1	74LS20	7	14			
U2	74LS05	7	14			
U3	74LS00	7	14			
U4	74LS245	10	20			
U5	74LS245	10	20			
U6	74LS241	10	20			
U7	74LS241	10	20			
U8	74LS373	10	20			
U9	74LS373	10	20			
U10	74C109	7	14			
U11	74C109	7	14			
U12	74LS175	8	16			
U13	74C109	7	14			
U14	74LS138	8	16			
U15	74LS20	7	14			
U16	74LS93	10	5			
U17	ADC1210	21	22	20		
U18	74LS74	7	14			
U19	LM555	1	8			
U20	74LS04	7	14			
U21	74121	7	14			
U22	MC1741				7	14
U23	MC14051B	8	16	7		
U24	7400	7	14			
U25	74LS175	8	16			
U26	LF398				1	4
U27	LF398	7			1	4
U28	LF398	7			1	4
U29	MC1741				7	4
U30	MC1741				7	4
U31	MC1741				7	4
U32	MC1741				7	4
U33	MC1741				7	4
U34	MC1741				7	4
U35	74LS04	7	14			
U36	74LS175	8	16			
U37	74LS138	8	16			
U38	DAC1210LCD	3,12		24		
U39	DAC1210LCD	3,12		24		
U40	DAC1210LCD	3,12		24		

TABLE A.2

Device Reference

Reference	Type	GND	+5V	-5V	+12V	-12V
U41	MC1741	3			7	4
U42	MC1741	3			7	4
U43	MC1741	3			7	4
U44	74LS74	7	14			
U45	74LS175	8	16			
U46	74LS93	10	5			
U47	8231A	1	2		16	
U48	74LS245	10	20			
U49	74LS04	7	14			
U50	74LS00	7	14			
U51	74LS00	7	14			
U52	74LS175	8	16			
U53	2764	14	1,27,28			
U54	2764	14	1,27,28			

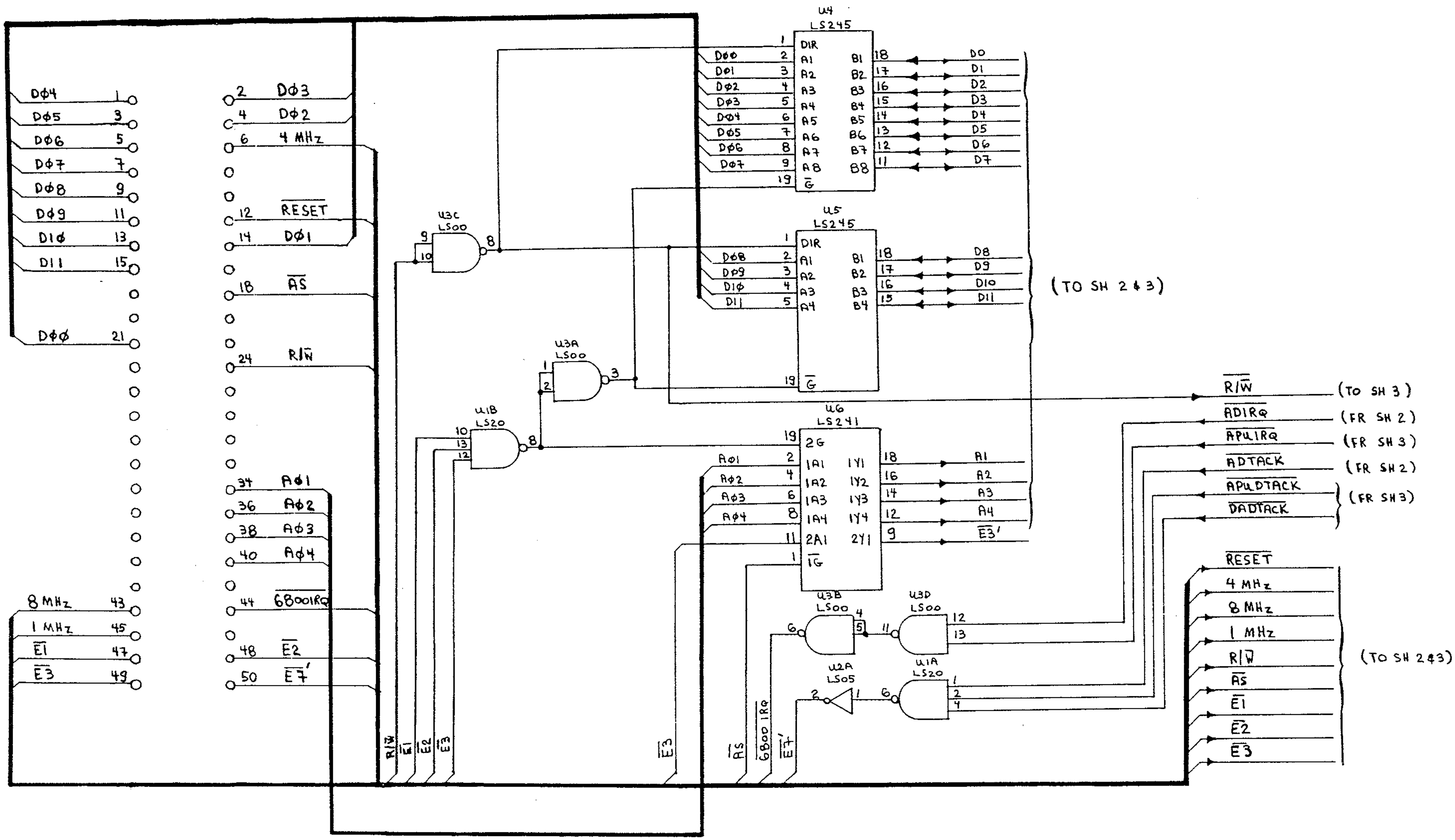


Figure A.1: SAAB Schematic Diagram (sheet 1 of 3)

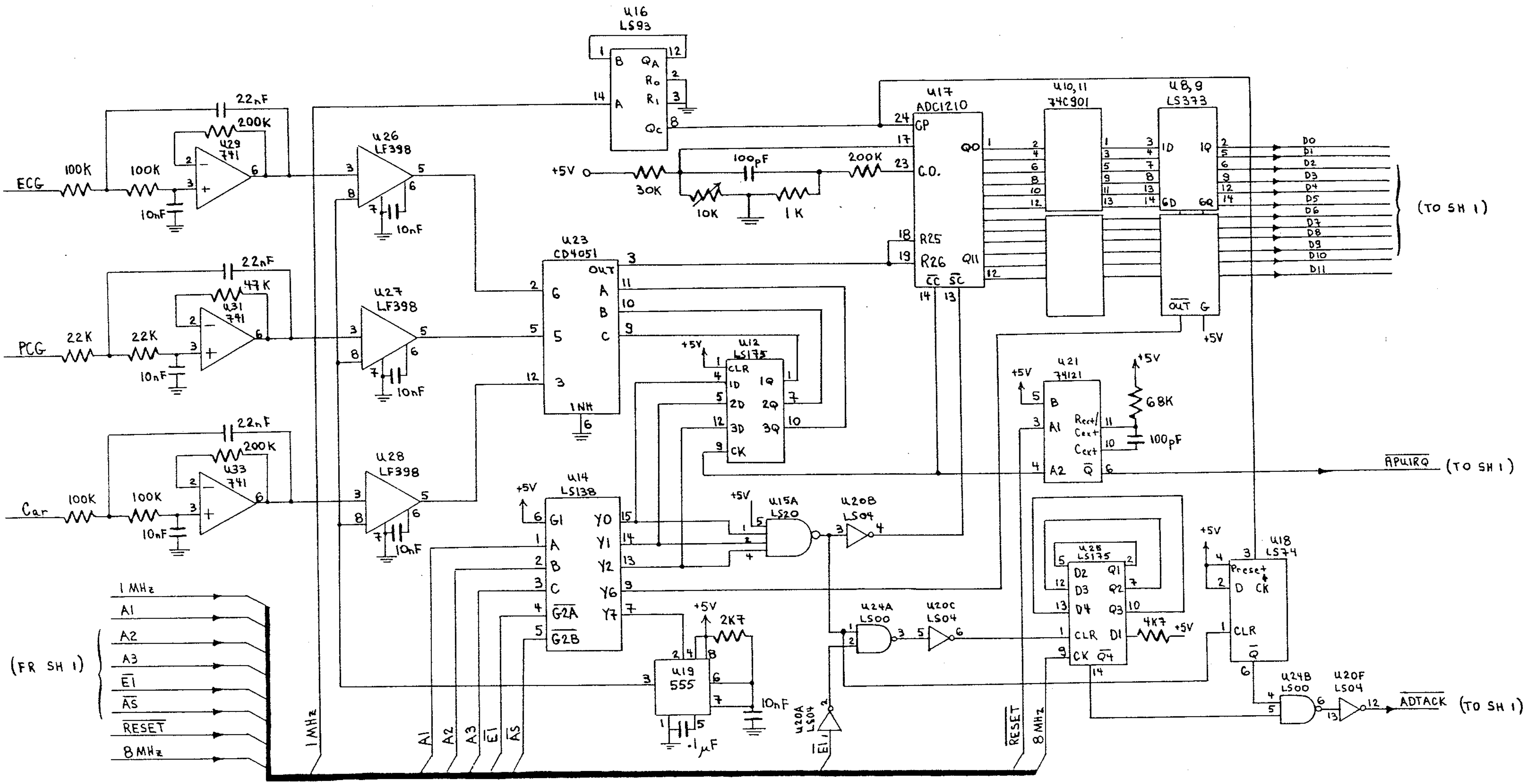


Figure A.2: SAAB Schematic Diagram (sheet 2 of 3)

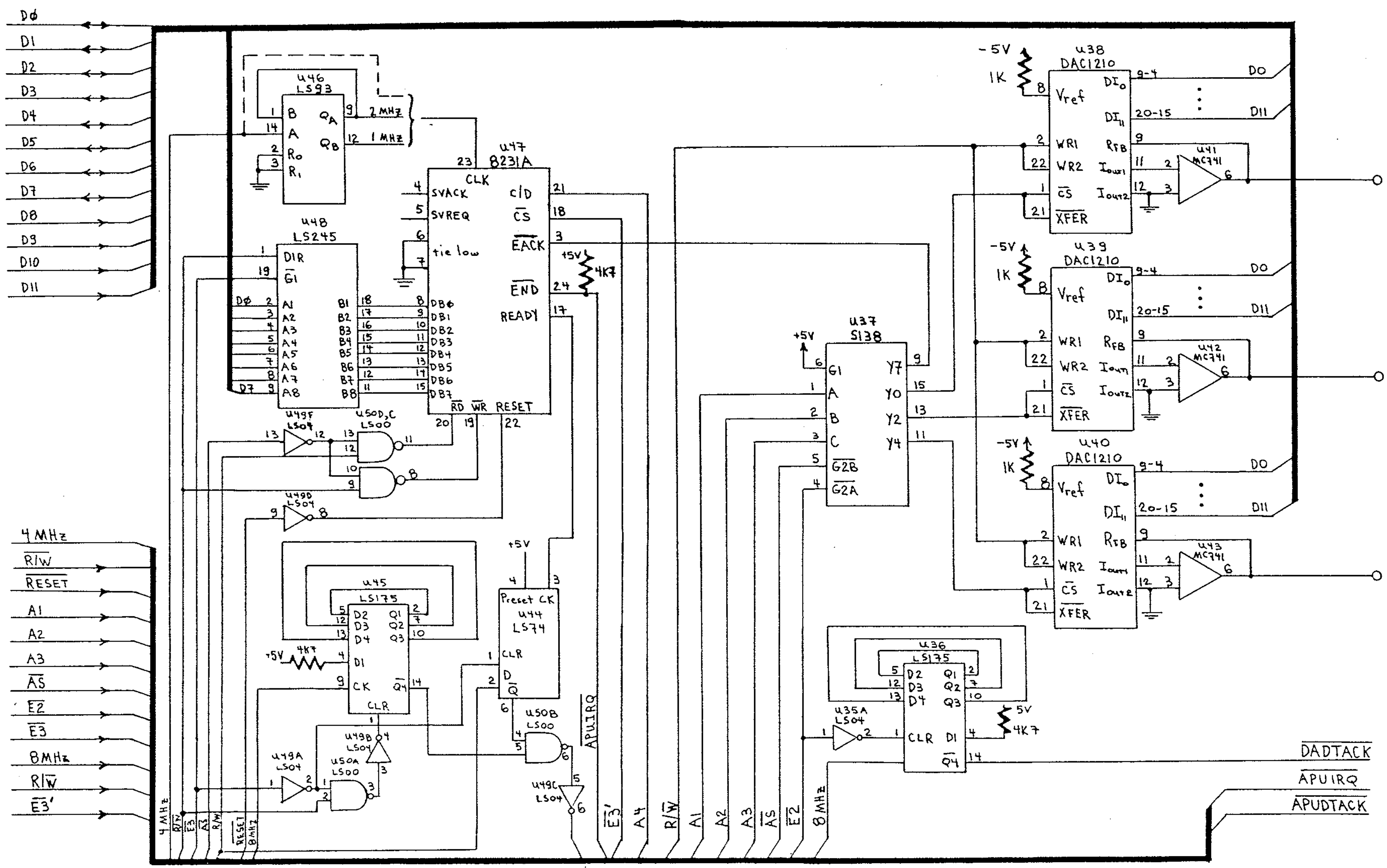


Figure A.3: SAAB Schematic Diagram (sheet 3 of 3)

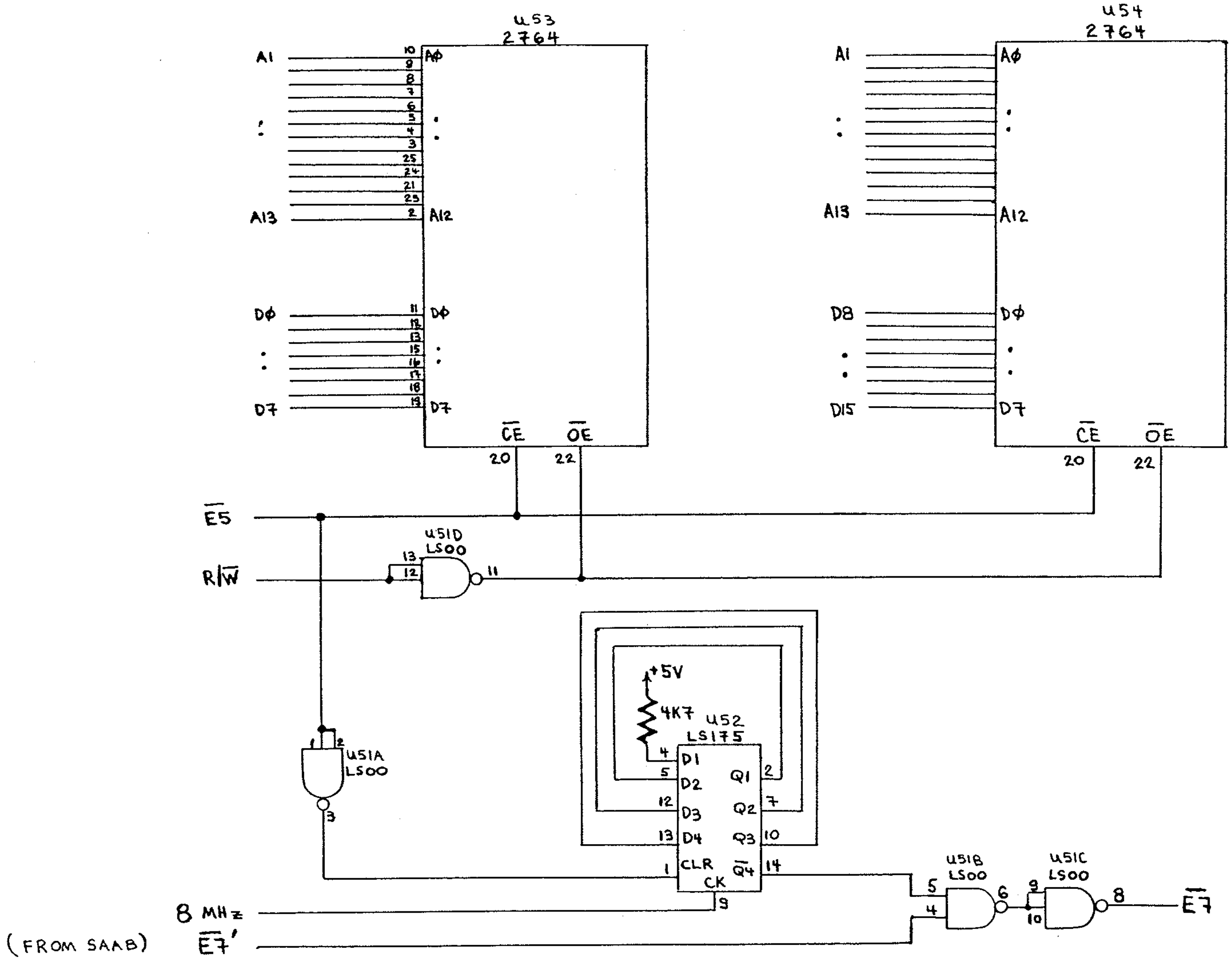


Figure A.4: USER ROM Schematic Diagram

Appendix B

SOFTWARE LISTING

```
                  * NAM THESIS PROJECT, YOU KNOW!!
00000900          MEMST    EQU   $0900 START OF RAM
00005FFE          MEMEND   EQU   $5FFE END OF RAM
00000900          ECGSTRT SET    $0900 STARTING ADDRESS OF ECG SAMPLES
00001500          ECGEND   SET   $1500 ENDING ADDRESS OF ECG SAMPLES
00001500          CARSTRT SET    $1500 STARTING ADDRESS OF PCG SAMPLES
00002100          CAREND   SET   $2100 ENDING ADDRESS OF PCG SIGNALS
00002100          PCGSTRT SET    $2100 STARTING ADDRESS OF CAROTID SAMPLES
00002D00          PCGEND   SET   $2D00 ENDING ADDRESS OF CAROTID SAMPLES
00001B00          TRECGST EQU    $1B00 STARTING ADDRESS OF TRANSFORMED ECG
000020FC          TRECGEN EQU    $20FC ENDING ADDRESS OF TRANSFORMED ECG
00002100          TRCARST EQU    $2100 STARTING ADDRESS OF TRANSFORMED CAROTID
00002300          TRCAREN EQU    $2300 ENDING ADDRESS OF TRANSFORMED CAROTID
00002300          PCGENST EQU    $2300 STARTING ADDRESS OF PCG ENERGY CURVE
00003700          TMPSTA   EQU   $3700 TEMPORARY STORAGE OF CONVOLVED SAMPLES
00003700          XRST     EQU   $3700 STARTING ADDRESS OF REAL PART OF PCG
00003EFC          XRENH    EQU   $3EFC HALFWAY POINT REAL ARRAY
00004700          XREN     EQU   $4700 ENDING ADDRESS OF REAL PART OF ECG
00004700          XIST     EQU   $4700 STARTING ADDRESS OF IMAGINARY PART OF PCG
00005700          XIEN     EQU   $5700 ENDING ADDRESS OF IMAGINARY PART OF PCG
00005700          TMPSTB   EQU   $5700 TEMPORARY STORAGE OF POWER SPECTRUM DATA
00005EFC          TMPENB   EQU   $5EFC END OF TEMPORARY STORAGE
00004700          TMPSTC   EQU   $4700 STARTING ADDRESS OF POWER SPECTRUM CONV.
                  *
                  *   I/O   DEVICE  ADDRESSES
                  *
00010021          TCR      EQU   $10021 PI/T TIMER CONTROL REGISTER
00010025          CPR      EQU   $10025 PI/T COUNTER PRELOAD REGISTER
0001002D          CNTR     EQU   $1002D PI/T COUNTER
00010035          TSR      EQU   $10035 PI/T TIMER STATUS REGISTER
00020001          ADCH1    EQU   $20001 A/D CHANNEL 1
00020003          ADCH2    EQU   $20003 A/D CHANNEL 2
00020005          ADCH3    EQU   $20005 A/D CHANNEL 3
0002000C          ADREAD   EQU   $2000C READ A/D CONVERTER
0002000E          HOLD     EQU   $2000E START SAMPLE-HOLDS
00040000          DACH1    EQU   $40000 D/A CHANNEL 1 WRITE AND LATCH
00040004          DACH2    EQU   $40004 D/A CHANNEL 2 WRITE AND LATCH
00040008          DACH3    EQU   $40008 D/A CHANNEL 3 WRITE AND LATCH
0004000E          APUEACK EQU    $4000E APU INTERRUPT ACKNOWLEDGE
00050001          APUOPER EQU    $50001 COMMAND TO W/R OPERANDS TO/FROM APU
00050011          APUCOM   EQU   $50011 SEND COMMAND TO APU
                  *
                  *   VARIOUS DATA REGISTERS USED BY ROUTINES
                  *
000000 00FF        DS.B 255             AREA OF MEMORY USED FOR DATA REGISTERS
00007F00          VDR      EQU    $7F00
00007F00          QRS1     EQU    VDR   NUMBER OF BYTES TO QRS1
00007F02          QRS2     EQU    VDR+2 NUMBER OF BYTES TO QRS2
00007F04          T2PEAK   EQU    VDR+4 NUMBER OF BYTES TO DICROTIC NOTCH USING
                  *                     T2 PEAK OF CAROTID TRANSFORM.
00007F06          DCRTC    EQU    VDR+6 NUMBER OF BYTES TO DICROTIC NOTCH USING
                  *                     CDCRTC ROUTINE.
00007F08          PCGS1A   EQU    VDR+8 NUMBER OF BYTES TO S1
00007F0A          PCGS1B   EQU   VDR+10 NUMBER OF BYTES TO SECOND S1 (QRS2)
00007F0C          PCGS2    EQU   VDR+12 NUMBER OF BYTES BETWEEN S1 AND S2
00007F0E          PCGENSE EQU    VDR+14 END OF SYSTOLIC PCG ENERGY CURVE
00007F10          PCGENDE EQU    VDR+16 END OF DIASTOLIC PCG ENERGY CURVE
00007F12          SEQNUM   EQU   VDR+18
00007F16          EDCTSYS EQU    VDR+22 TIME DOMAIN EDC FOR SYSTOLIC SEGMENT
00007F18          EDCTDIA EQU    VDR+24 TIME DOMAIN EDC FOR DIASTOLIC SEGMENT
00007F1A          EDCFSYS EQU    VDR+26 FREQ DOMAIN EDC FOR SYSTOLIC SEGMENT
```

```
00007F1E      EDCFDIA EQU   VDR+30 FREQ DOMAIN EDC FOR DIASTOLIC SEGMENT
00007F2A      L       EQU   VDR+42
00007F2C      LE      EQU   VDR+44
00007F2E      LE1     EQU   VDR+46
00007F30      I       EQU   VDR+48
00007F32      J       EQU   VDR+50
00007F34      WR      EQU   VDR+52
00007F38      WI      EQU   VDR+56
00007F3C      UR      EQU   VDR+60
00007F40      UI      EQU   VDR+64
00007F44      QUOT    EQU   VDR+68
00007F48      COUNTER EQU   VDR+72
              *
              *
              *   A/D AND APU CONSTANTS, COMMANDS, AND REGISTERS
              *
              *
0000007A      ADSMPL  EQU      122 COUNTER VALUE - 1024 HZ SAMPLING RATE
000001E8      ADWNSMP EQU      488 COUNTER VALUE - 256 HZ SAMPLING RATE
00000001      STRTIMR EQU      $01 START TIMER
00000000      HLTIMR  EQU      $00 HALT TIMER
00000070      ADEXADD EQU      $70 A/D EXCEPTION ADDRESS
00000070      APUEXAD EQU      $70 APU EXCEPTION ADDRESS
              *
              *
              *  ECG, CAROTID, PCG  ADDRESSES AND FFT CONSTANTS
              *
              *
00000900      ECGSNEW EQU    $0900 NEW ECG START ADDRESS
00000C00      ECGENEW EQU    $0C00 NEW ECG END ADDRESS
00000C00      CARSNEW EQU    $0C00 NEW CAROTID START ADDRESS
00000F00      CARENEW EQU    $0F00 NEW CAROTID END ADDRESS
00000F00      PCGSNEW EQU    $0F00 NEW PCG START ADDRESS
00001B00      PCGENEW EQU    $1B00 NEW PCG END ADDRESS
00000008      ECGDWN  EQU      $08 ECG DOWNSAMPLE FACTOR (/2)
00000008      CARDWN  EQU      $08 CAROTID DOWNSAMPLE FACTOR
00000002      PCGDWN  EQU      $02 PCG DOWNSAMPLE FACTOR (/2)
00000010      ECGCNV  EQU       16 ECG CONVOLVE WINDOW WIDTH
00000010      CARCNV  EQU       16 CAR CONVOLVE WINDOW WIDTH
00000020      PCGCNV  EQU       32 PCG CONVOLVE WINDOW WIDTH
00000098      T1TOT2  EQU      152 MOVE AHEAD THIS MANY BYTES TO LOOK FOR T2
0000001B      DS2DEL  EQU       27 DELAY BETWEEN S2 AND DICROTIC NOTCH (53 MS
0000000A      LNN     EQU       10
00000400      N       EQU     1024 N=2**LNN
00000200      NV2     EQU      N/2 NV2=N/2
000003FF      NM1     EQU      N-1 NM1=N-1
008A3D70      HAM54 EQU $008A3D70 .54 FLOATING POINT
7FEB851E      HAM46 EQU $7FEB851E .46 FLOATING POINT
              *
              *
              * APU COMMANDS
              *
              *
00000006      ACOS    EQU $06    32-BIT FLOATING-POINT INVERSE COSINE
00000005      ASIN    EQU $05    32-BIT FLOATING-POINT INVERSE SINE
00000007      ATAN    EQU $07    32-BIT FLOATING-POINT INVERSE TANGENT
00000034      CHSD    EQU $34    32-BIT FIXED-POINT SIGN CHANGE
00000015      CHSF    EQU $15    32-BIT FLOATING-POINT SIGN CHANGE
00000074      CHSS    EQU $74    16-BIT FIXED-POINT SIGN CHANGE
00000003      COS     EQU $03    32-BIT FLOATING-POINT COSINE
0000002C      DADD    EQU $2C    32-BIT FIXED-POINT ADD
```

```
0000002F    DDIV     EQU $2F   32-BIT FIXED-POINT DIVIDE
0000002E    DMUL     EQU $2E   32-BIT FIXED-POINT MULTIPLY, LOWER
00000036    DMUU     EQU $36   32-BIT FIXED-POINT MULTIPLY, UPPER
0000002D    DSUB     EQU $2D   32-BIT FIXED-POINT SUBTRACT
0000000A    EXP      EQU $0A   32-BIT FLOATING POINT EXP(X)
00000010    FADD     EQU $10   32-BIT FLOATING POINT ADD
00000013    FDIV     EQU $13   32-BIT FLOATING-POINT DIVIDE
0000001E    FIXD     EQU $1E   32-BIT FLOATING-POINT TO 32-BIT FIXED POINT
0000001F    FIXS     EQU $1F   32-BIT FLOATING-POINT TO 16-BIT FIXED-POINT
0000001C    FLTD     EQU $1C   32-BIT FIXED-POINT TO 32-BIT FLOATING-POINT
0000001D    FLTS     EQU $1D   16-BIT FIXED-POINT TO 32-BIT FLOATING-POINT
00000012    FMUL     EQU $12   32-BIT FLOATING-POINT MULTIPLY
00000011    FSUB     EQU $11   32-BIT FLOATING-POINT SUBTRACT
00000008    LOG      EQU $08   32-BIT FLOATING-POINT COMMON LOG
00000009    LN       EQU $09   32-BIT FLOATING-POINT NATURAL LOG
00000000    NOPAPU   EQU $00   NO OPERATION
00000038    POPD     EQU $38   32-BIT STACK POP
00000018    POPF     EQU $18   32-BIT STACK POP (SAME AS POPD)
00000078    POPS     EQU $78   16-BIT STACK POP
00000037    PTOD     EQU $37   PUSK 32-BIT TOS ONTO STACK
00000017    PTOF     EQU $17   PUSH 32-BIT TOS ONTO STACK (SAME AS PTOD)
00000077    PTOS     EQU $77   PUSH 16-BIT TOS ONTO STACK
0000001A    PUPI     EQU $1A   PUSH 32-BIT FLOATING-POINT PI
0000000B    PWR      EQU $0B   32-BIT FLOATING-POINT X(Y)
0000006C    SADD     EQU $6C   16-BIT FIXED-POINT ADD
0000006F    SDIV     EQU $6F   16-BIT FIXED-POINT DIVIDE
00000002    SIN      EQU $02   32-BIT FLOATING-POINT SINE
0000006E    SMUL     EQU $6E   16-BIT FIXED-POINT MULTIPLY,LOWER
00000076    SMUU     EQU $76   16-BIT FIXED-POINT MULTIPLY,UPPER
00000001    SQRT     EQU $01   32-BIT FLOATING-POINT SQUARE ROOT
0000006D    SSUB     EQU $6D   16-BIT FIXED-POINT SUBTRACT
00000004    TAN      EQU $04   32-BIT FLOATING-POINT TANGENT
00000039    XCHD     EQU $39   EXCHANGE 32-BIT STACK OPERANDS
00000019    XCHF     EQU $19   EXCHANGE 32-BIT STACK OPERANDS (SAME AS XCHD
00000079    XCHS     EQU $79   EXCHANGE 16-BIT STACK OPERANDS
            *
            *
            *        TRAP 14 FUNCTION CODES
            *
            *
000000FD    LINKIT   EQU 253     APPEND USER TABLE TO TRAP 14 TABLE
000000F8    OUTCH    EQU 248     OUTPUT SINGLE CHARACTER TO PORT 1
000000F7    INCHE    EQU 247     INPUT SINGLE CHARACTER FROM PORT 1
000000F3    OUTPUT   EQU 243     OUTPUT STRING TO PORT1
000000EC    HEX2DEC  EQU 236     CONVERT HEX TO ASCII
000000EA    PUTHEX   EQU 234     CONVERT 1 HEX DIGIT --> ASCII
000000E9    PNT2HX   EQU 233        "    2  "    "    -->   "
000000E8    PNT4HX   EQU 232        "    4  "    "    -->   "
000000E7    PNT6HX   EQU 231        "    6  "    "    -->   "
000000E6    PNT8HX   EQU 230        "    8  "    "    -->   "
000000E5    START    EQU 229     RESTART TUTOR
000000E4    TUTOR    EQU 228     GO TO TUTOR
000000E3    OUT1CR   EQU 227     OUTPUT STRING PLUS 'CR','LF' TO PORT 1
000000E2    GETNUMA  EQU 226     CONVERT ASCII ENCODED HEX TO HEX
000000E1    GETNUMD  EQU 225     CONVERT ASCII CODED DECIMAL TO HEX
00000000    ERROR1   EQU  0      ERROR #1 MESSAGE: QRS COMPLEXES
            *
```

```
*##############################################################
*                       MACRO LIST
*##############################################################
*
CALLCNVL MACRO
*
* SETS POINTER FOR AND CALLS CONVOLVE ROUTINE FOR ECG AND
* CAROTID TRANSFORMS AND PCG ENERGY CURVE.
* MACRO CALL IS 'TRANSFORM START,CONVOLVE FACTOR,TRANSFORM END
*
 MOVE.W #\1,A0          STARTING ADDRESS OF TRANSFORMED SIGNAL
 MOVE.W #\2,D0          CONVOVLE WINDOW WIDTH
 MOVE.W #\3,A1          ENDING ADDRESS OF TRANSFORMED SIGNAL
 JSR CNVLV              GOTO CONVOLVE SUBROUTINE
 ENDM
*##############################################################
CLDATA2 MACRO
*
* CLEARS TWO DATA REGISTERS
*
 CLR.\0 \1
 CLR.\0 \2
 ENDM
*##############################################################
CLDATA3 MACRO
*
* CLEARS THREE DATA REGISTERS
*
 CLR.\0 \1
 CLR.\0 \2
 CLR.\0 \3
 ENDM
*##############################################################
CLDATA8 MACRO
*
* CLEARS EIGHT DATA REGISTERS
*
 CLR.\0 D0
 CLR.\0 D1
 CLR.\0 D2
 CLR.\0 D3
 CLR.\0 D4
 CLR.\0 D5
 CLR.\0 D6
 CLR.\0 D7
 ENDM
*##############################################################
CONVERT MACRO
*
* USED FOR A/D CONVERSION. DUMMY WRITE TO TIE PROPER MUX INPUT
* CHANNEL TO OUTPUT CHANNEL AND TO START A/D CONVERTER. THEN
* WAIT FOR INTERRUPT. AFTER INTERRUPT INCREMENT CHANNEL COUNTE
* MACRO CALL IS 'A/D CHANNEL EA,COUNTER DATA REG'
*
 MOVE.B #0,\1             MUX IN<-->OUT AND START A/D
 STOP   #$2300            WAIT UNTIL CONVERSION IS FINISHED.
*                         PUT PROCESSOR IN SUPERVISOR STATE WITH
*                         INTERRUPT PRIORITY LEVEL 0.
 ADDI.B #1,\2             INCREMENT A/D CHANNEL COUNTER
 ENDM
*##############################################################
```

```
           DATAMV MACRO
           *
           * MOVES A BLOCK OF DATA FROM ONE LOCATION TO ANOTHER W/DOWNSAM
           * MACRO CALL IS 'ORIGINAL STARTING ADDRESS,ADDRESS REGISTER,
           * ORIGINAL END ADDRESS,ADDRESS REGISTER,NEW START,NEW END,
           * ADDRESS REGISTER,DOWNSAMPLE FACTOR'
           *
            STPOIN2 \1,\2,\3,\4       OLD ECG DATA POINTER
           \1 SET \5
           \3 SET \6                  SET NEW START AND END ADDRESSES
            MOVE.W #\1,\7             NEW DATA POINTER (DOWNSAMPLED DATA)
           \@ MOVE.W (\2),(\7)+       MOVE DATA
            ADD.W #\8,\2              DOWNSAMPLE
            CMP.W \4,\2
            BLT.S \@                  CONTINUE UNTIL ALL DATA MOVED
            ENDM
           *############################################################
           DATIMER MACRO
           *
           * INITIALIZE PI/T REGISTERS AND START COUNTING. ALL PARAMETERS
           * HAVE BEEN SET UP FOR D/A ROUTINES. NEED SAMPLING PERIOD IN
           * D0.
           *
            STPOIN2.L CNTR,A5,CPR,A6  COUNTER,PRELOAD ADDRESS
            MOVE.L D0,D7               SAMPLING PERIOD; NOTE THAT THERE IS
           *                           PRESCALER (/32); WITH THE CLK=4MHZ,
           *                           A COUNTER VALUE OF 1 EQUALS 8 US.
            MOVEP.L D7,0(A6)           SAMPLING PERIOD --> PRELOAD REGISTE
            MOVE.B #1,TSR              RESET TSR
            MOVE.B #STRTIMR,TCR        START TIMER USING TCR RUN CODE
            ENDM
           *############################################################
           DELAY MACRO
           *
           * CAUSES A TIME DELAY. A DELAY VALUE OF '1' CORRESPONDS TO
           * APPROXIMATELY 4.5 USEC.
           * MACRO CALL IS 'DELAY VALUE IM,DATA REGISTER'
           *
            MOVE.W #\1,\2
           \@ DBRA \2,\@
            ENDM
           *############################################################
           DSPLOPT MACRO
           *
           * DISPLAY D/A OPTIONS ON TERMINAL.
           * MACRO CALL IS 'START ADDRESS OF MESSAGE EA,END ADDRESS EA'
           *
            LEA \1,A5
            LEA \2,A6
            TRP14 OUT1CR
            ENDM
           *############################################################
           GETADD MACRO
           *
           *  COMPUTES ADDRESS FROM NUMBER OF BYTES AND STARTING ADDRESS.
           *  MACRO CALL IS 'NUMBER OF BYTES EA,ADDRESS REGISTER,STARTING
           *  ADDRESS'
           *
            MOVE.W \1,\2
            ADD.W #\3,\2
            ENDM
```

```
*##############################################################
ML3216 MACRO
*
* THIS MACRO MULTIPLIES A 32 BIT NUMBER BY A 16 BIT NUMBER.
* RESULT (48 BITS) IS RETURNED IN MULTIPLICAND (LOWER 32 BITS)
* AND TEMP DR2 (UPPER 16 BITS).
* MACRO CALL IS 'MULTIPLIER (16 BITS) EA,MULTIPLICAND (32 BITS
* TEMP DR1,TEMP DR2,TEMP DR3'
*
 CLR.L \4
 MOVE.W \2,\3            LOWER WORD OF MULTIPLICAND INTO TEMP
 MULU.W \1,\3            MULTIPLY TEMP BY MULTIPLIER
 SWAP \2                 UPPER WORD OF MULTIPLICAND INTO LOWER WO
 MULU.W \1,\2            MULTIPLY LOWER WORD BY MULTIPLIER
 MOVE.B #15,\5
\@ LSL.L #1,\2
 ROXL.L #1,\4
 DBRA \5,\@
 ADD.L  \3,\2            ADD TO GET LOWER 32 BITS
 CLR.L \5
 ADDX.L \5,\4            ADD TO GET UPPER 16 BITS
 ENDM
*##############################################################
SPACE MACRO
*
* OUTPUTS BLANK LINE TO TERMINAL.
*
 LEA SPAC,A5
 LEA SPCE,A6
 TRP14 OUT1CR
 ENDM
*##############################################################
STPOIN2 MACRO
*
* SETS TWO POINTERS. MUST INDICATE WHETHER B,W,LW.
* MACRO CALL IS 'VALUE IM,ADDRESS REGISTER' * 2
*
 MOVE.\0 #\1,\2
 MOVE.\0 #\3,\4
 ENDM
*##############################################################
STPOIN3 MACRO
*
* SETS THREE POINTERS. MUST INDICATE WHETHER B,W,LW.
* MACRO CALL IS 'VALUE IM,ADDRESS REGISTER' * 3
*
 MOVE.\0 #\1,\2
 MOVE.\0 #\3,\4
 MOVE.\0 #\5,\6
 ENDM
*##############################################################
SQUARE MACRO
*
* THIS MACRO SQUARES A WORD AND STORES IT AS A LONG WORD.
* MACRO CALL IS 'DATA REGISTER (WORD),DATA REGISTER (LW),
* ADDRESS REGISTER (MEMORY POINTER)'
*
 MOVE.W \1,\2
 MULS.W \1,\2            SQUARE NUMBER
 MOVE.L \2,(\3)+         STORE QUANTITY AS LONG WORD
 ENDM
```

```
*##############################################################
TEMPMOVE MACRO
*
* MOVES DATA FROM TEMPORARY MEMORY (FROM CONVOLVE S/R) TO
* ACTUAL TRANSFORM MEMORY.
* MACRO CALL IS 'TRANSFROM START,MEMORY POINTER'
*
 STPOIN2 \1,A0,TMPSTA,A1
\@ MOVE.L (A1)+,(A0)+      MOVE FROM TEMP TO TRANSFORM
 CMP.W #\2,A0
 BLT \@
 ENDM
*##############################################################
TIMER MACRO
*
* INITIALIZE PI/T REGISTERS AND START COUNTING.
* MACRO CALL IS 'COUNTER ADDRESS REGISTER,PRELOAD ADDRESS
* REGISTER,SAMPLING PERIOD,PERIOD DATA REGISTER'
*
 STPOIN2.L CNTR,\1,CPR,\2  COUNTER,PRELOAD ADDRESS
 MOVE.L #\3,\4             SAMPLING PERIOD; NOTE THAT THERE IS
*                          PRESCALER (/32); WITH THE CLK=4MHZ,
*                          A COUNTER VALUE OF 1 EQUALS 8 US.
 MOVEP.L \4,0(\2)          SAMPLING PERIOD --> PRELOAD REGISTE
 MOVE.B #1,TSR             RESET TSR
 MOVE.B #STRTIMR,TCR       START TIMER USING TCR RUN CODE
 ENDM
*##############################################################
TRP14 MACRO
*
* USED TO INVOKE TUTOR TRAP 14 HANDLER.
* MACRO CALL IS 'FUNCTION NUMBER IM'.
*
 MOVE.B #\1,D7
 TRAP #14
 ENDM
*##############################################################
TRUNCAT MACRO
*
*  THIS MACRO TRUNCATES A 32 BIT LONG WORD TO 12 BITS.
*  MACRO CALL IS 'SIGNAL POINTER REGISTER,DATA REGISTER'
*
 MOVE.L (\1)+,\2
 SWAP \2
 LSR.W #4,\2
 ENDM
*##############################################################
TSTIMR MACRO
*
* TEST TSR TO DETERMINE IF TIMER HAS COUNTED DOWN TO ZERO.
* AFTER COUNTING DOWN DETERMINE IF ALL DATA HAS BEEN PROCESSED
* MACRO CALL IS 'END ADDRESS REGISTER,SIGNAL POINTER REGISTER'
*
\@ CMP.B #0,TSR
 BEQ \@
 MOVE.B #1,TSR
 CMP.W \1,\2
 ENDM
*
```

```
*###########################################################
*###########################################################
*#                                                         #
*#                                                         #
*#                APU  &  FFT  MACROS                      #
*#                                                         #
*#                                                         #
*###########################################################
*###########################################################
*
*
APU MACRO
*
*  ISSUE A COMMAND TO MANIPULATE DATA ON APU STACK.
*  MACRO CALL IS 'COMMAND IM'
*
 MOVE.B #\1,APUCOM
 STOP #$2300
 ENDM
*###########################################################
APUSGRD MACRO
*
*  READ 32-BIT WORD FROM APU TO SIGNAL ARRAY
*  MACRO CALL 'SIGNAL AR,ARRAY SCRIPT DR,APU ADDRESS AR'
*
 MOVE.W #$3,D7
\@ ROL.L #$8,D6
 MOVE.B (\3),D6
 DBRA D7,\@
 MOVE.L D6,0(\1,\2)
 ENDM
*###########################################################
APUSGWR MACRO
*
*  MOVE A 32-BIT NUMBER FROM SIGNAL ARRAY TO APU
*  MACRO CALL IS 'SIGNAL POINTER AR,ARRAY SCRIPT DR,APU ADDRES
*
 MOVE.L 0(\1,\2),D6
 MOVE.W #$3,D7
\@ MOVE.B D6,(\3)
 ROR.L #$8,D6
 DBRA D7,\@
 ENDM
*###########################################################
APURD MACRO
*
*   READ 32-BIT NUMBER FROM APU.
*   MACRO CALL IS 'APU OPERAND ADDRESS AR,DATA DR'
*
 MOVE.W #$3,D7
\@ ROL.L #$8,\2
 MOVE.B (\1),\2
 DBRA D7,\@
 ENDM
*###########################################################
APUWR MACRO
*
*  MOVE A 32-BIT NUMBER FROM MEMORY TO APU
*  MACRO CALL IS 'VALUE DR,APU ADDRESS AR'
*
 MOVE.W #$3,D7
```

```
\@ MOVE.B \1,(\2)
 ROR.L #$8,\1
 DBRA D7,\@
 ENDM
*################################################################
APUWR16 MACRO
*
*  MOVE A 16-BIT NUMBER FROM MEMORY TO APU
*  MACRO CALL IS 'VALUE DR,APU ADDRESS AR'
*
 MOVE.W #$1,D7
\@ MOVE.B \1,(\2)
 ROR.L #$8,\1
 DBRA D7,\@
 ROR.L #$8,\1
 ROR.L #$8,\1
 ENDM
*################################################################
POW2TOA MACRO
*
* CALCULATES 2**A. MACRO CALL IS 'A EA'
*
 MOVE.W \1,D7
 SUBI.W #1,D7
 MOVE.W #1,D6
\@ MULU.W #2,D6
 DBRA D7,\@
 ENDM
*################################################################
REORDER MACRO
*
*  MACRO CALL IS ADDRESS AR,J DR,T DR,I DR
*
 MOVE.L 0(\1,\2),\3
 MOVE.L 0(\1,\4),0(\1,\2)
 MOVE.L \3,0(\1,\4)
 ENDM
*
*
```

```
                    *##################################################################
                    *                            MAINLINE
                    *##################################################################
                    *      MAINLINE CALLS THE FOLLOWING SUBROUTINES:
                    *            USRFNCS:   SET UP USER TRAP 14 TABLE
                    *            ADCONV:    PERFORM A/D CONVERSION
                    *            ECG:       PERFORM ECG PROCESSING
                    *            CAR:       PERFORM CAROTID PROCESSING
                    *            PCGT:      PERFORM PCG PROCESSING (TIME DOMAIN)
                    *            PCGF:      PERFORM PCG PROCESSING (FREQ DOAMIN)
                    *            DAFIX:     DISPLAY SIGNALS USING D/A CONVERTERS
                    *            FIN:       END PROGRAM
                    *##################################################################
                    *
00006000             ORG $006000
006000 4280          CLR.L D0
006002 307C0900             MOVE.W #MEMST,A0
006006 20C0   MEMCLR MOVE.L D0,(A0)+      CLEAR MEMORY
006008 B0FC5FFE             CMP.W #MEMEND,A0
00600C 6FF8          BLE.S MEMCLR
00600E 343C001E             MOVE.W #30,D2
006012 4EB872D4      MAIN JSR BLANK            CLEAR SCREEN
006016 51CAFFFA             DBRA D2,MAIN
00601A 4EB860CE             JSR USRFNCS
00601E 4EB86126             JSR ADCONV
006022 21FC00006240
       0070          MOVE.L #APISR,APUEXAD
00602A 4EB8624A             JSR ECG
00602E 4EB863CA             JSR CAR
006032 4EB865A0             JSR PCGT
006036 4EB86F02             JSR PCGF
00603A 4EB872D4             JSR BLANK
                     DSPLOPT QUIT,QUITE
00603E 4BF86064             LEA QUIT,A5
006042 4DF8608C             LEA QUITE,A6
                     TRP14 OUT1CR
006046 1E3C00E3             MOVE.B #OUT1CR,D7
00604A 4E4E          TRAP #14
                     TRP14 INCHE
00604C 1E3C00F7             MOVE.B #INCHE,D7
006050 4E4E          TRAP #14
006052 0C000051             CMP.B #$51,D0
006056 6704          BEQ.S GOFIN
006058 4EB87206             JSR DAFIX
00605C 4EB872D4      GOFIN JSR BLANK
006060 4EB87A8E             JSR FIN
006064 4F     QUIT DC.W 'OPTION: QUIT (Q) OR DISPLAY RESULTS <R>'
00608C 20     QUITE DC.W ' '
              *
              *
```

```
                *************************************************************
                *                   CONVOLVE SUBROUTINE
                *·    CALLED BY:  ECNVL, CCNVL, PTCNVS, PTCNVD
                *************************************************************
                *      A0 ==> A2 <--> TRANSFORMED SIGNAL POINTER
                *      A1 <--> END OF TRANSFORMED SIGNAL
                *      A3 <--> CONVOLVE POINTER
                *      A4 <--> TEMPORARY RESULT POINTER
                *      D0 <--> CONVOLVE WINDOW WIDTH (M)
                *      D1 <--> C(N) = SUMMATION .....
                *      D2 <--> WINDOW WIDTH DOWN COUNTER
                *      D3 <--> TRANSFORM VALUE
                *      D4,D5,D6  <--> TEMPORARY DATA STORAGE
                *************************************************************
                *      CALL FOLLOWING MACROS:
                *           ML3216
                *************************************************************
                *   CONVOLVE SUBROUTINE USED BY ECG AND CAROTID TO SMOOTH ECG
                *   AND CAROTID TRANSFORM. USED BY PCG TO SMOOTH THE ENERGY CUR
                *   SET POINTERS. BLOCK OF MEMORY STARTING AT TMPSTA IS USED TO
                *   STORE THE CONVOLVED SIGNAL.
                *
00608E 3448     CNVLV MOVE.W A0,A2             TRANSFORMED SIGNAL POINTER
006090 387C3700       MOVE.W #TMPSTA,A4        TEMPORARY MEMORY POINTER
006094 584A      ADDQ.W #4,A2                  INCREMENT BY ONE LONG WORD
                *
                *   SET POINTERS FOR INNER LOOP CALCULATIONS
                *
006096 364A     CNVLV1 MOVE.W A2,A3            CONVOLVE INNER LOOP POINTER
006098 4281      CLR.L D1                      C(N)=0
00609A 3400      MOVE.W D0,D2                  NUMBER OF SAMPLES IN WINDOW
                *
                *   COMPUTE C(N) FOR SPECIFIC N ,I.E., C(N)=SUMMATION.
                *
00609C 2623     CNVLV2 MOVE.L -(A3),D3         DECREMENT, THEN LOAD DATA
00609E B6C8      CMP.W A0,A3
0060A0 6D22      BLT.S CNVLV3                  ENSURE THAT THE WINDOW DOES NOT
                *                              EXTEND BEFORE THE START POINTER.
                 ML3216 D2,D3,D4,D5,D6         X(N-K)W(K)
0060A2 4285      CLR.L D5
0060A4 3803      MOVE.W D3,D4
0060A6 C8C2      MULU.W D2,D4
0060A8 4843      SWAP D3
0060AA C6C2      MULU.W D2,D3
0060AC 1C3C000F       MOVE.B #15,D6
0060B0 E38B      @003 LSL.L #1,D3
0060B2 E395      ROXL.L #1,D5
0060B4 51CEFFFA       DBRA D6,@003
0060B8 D684      ADD.L D4,D3
0060BA 4286      CLR.L D6
0060BC DB86      ADDX.L D6,D5
0060BE D283      ADD.L D3,D1                   C(N) SUMMATION
0060C0 51CAFFDA       DBRA D2,CNVLV2           CONTINUE UNTIL ENTIRE WINDOW COMPUT
0060C4 28C1     CNVLV3 MOVE.L D1,(A4)+         C(N) COMPUTED FOR SPECIFIC N
0060C6 584A      ADDQ.W #4,A2
0060C8 B4C9      CMP.W A1,A2
0060CA 6DCA      BLT.S CNVLV1                  COMPUTE C(N) FOR ALL N
0060CC 4E75      RTS
                *
                *
```

```
                    *#################################################################
                    *#################################################################
                    *
                    *               DEFINE TABLE FOR TRAP 14 USER FUNCTIONS
                    *
                    *#################################################################
                    *
                    *      AO <--> START OF NEW TABLE, AFTER TRP14 A0 CONTAINS
                    *              THE POINTER TO THE OLD TABLE
                    *
                    *#################################################################
                    *
                    *      CALL FOLLOWING MACROS:
                    *           TRP14
                    *
                    *#################################################################
                    *#################################################################
                    *
                    *
                    *   THIS SUBROUTINE SETS UP THE LOOKUP TABLE OF THE STARTING
                    *   ADDRESSES OF USER-DEFINED FUNCTIONS CALLED BY THE TUTOR TRA
                    *   14 HANDLER. THE FORMAT FOR ENTRIES IN THIS TABLE IS $UUSSSS
                    *   WHERE $UU IS THE FUNCTION NUMBER AND $SSSSSS IS THE STARTIN
                    *   ADDRESS OF THE FUCNTION.
                    *
                    *
0060CE 41F860DE          USRFNCS LEA NEWTBL,A0       REGISTER A0 POINTS TO NEW TA
                     TRP14 LINKIT
0060D2 1E3C00FD            MOVE.B #LINKIT,D7
0060D6 4E4E          TRAP #14
0060D8 21C860E2            MOVE.L A0,ENDTBL    A0 POINTS TO OLD TABLE
0060DC 4E75         RTS
0060DE 0000         NEWTBL DC.W $0000
0060E0 60E6          DC.W UTRER1                   ERROR #1-QRS PEAK DETECT ERROR
0060E2 0000         ENDTBL DC.W $0000
0060E4 0000          DC.W $0000
                    *
                    *#################################################################
                    *                    ERROR HANDLING ROUTINES
                    *                    CALLED BY TRAP 14
                    *#################################################################
                    *
                    *   ERROR #1:  UNSUCCESSFUL ATTEMPT AT FINDING TWO CONSECUTIVE
                    *              QRS PEAKS. OUTPUT ERROR MESSAGE TO TERMINAL, THE
                    *              GO TO TUTOR.
                    *
0060E6 4EB872D4          UTRER1 JSR BLANK
                     DSPLOPT E1,E1E                OUTPUT ERROR MESSAGE
0060EA 4BF86100            LEA E1,A5
0060EE 4DF86124            LEA E1E,A6
                     TRP14 OUT1CR
0060F2 1E3C00E3            MOVE.B #OUT1CR,D7
0060F6 4E4E          TRAP #14
                     TRP14 TUTOR                   CONTROL TO TUTOR
0060F8 1E3C00E4            MOVE.B #TUTOR,D7
0060FC 4E4E          TRAP #14
0060FE 4E75          RTS
006100 45           E1 DC.W 'ERROR #1: 2 QRS COMPLEXES NOT FOUND'
006124 20           E1E DC.W ' '
                    *
```

```
                *###############################################################
                *###############################################################
                *
                *                       A/D  CONVERSION
                *
                *###############################################################
                *
                *      A0 <--> ECG POINTER
                *      A1 <--> PCG POINTER
                *      A2 <--> CAROTID POINTER
                *      A3 <--> PI/T COUNTER ADDRESS
                *      A4 <--> PI/T PRELOAD ADDRESS
                *      A5,A6 <--> USED BY TRAP 14 HANDLER
                *      D0 <--> SAMPLING PERIOD
                *      D1 <--> S/H DELAY
                *      D2 <--> A/D CHANNEL COUNTER REGISTER
                *
                *###############################################################
                *
                *      CALLS FOLLOWING MACROS:
                *            CONVERT '''      DELAY
                *            STPOIN3          TIMER->STPOIN2
                *            TRP14 ''
                *
                *###############################################################
                *###############################################################
                *
                *
                *   SET ECG, PCG, CAROTID POINTERS AND ISR, COUNTER, PRELOAD
                *   ADDRESSES. NOTE: INTERRUPTING DEVICE IS THE M6800 TYPE IRQ
                *   ISR IS AUTOVECTOR 28.
                *
                *
                ADCONV DSPLOPT ADST,ADSTE  DISPLAY START OF A/D
006126 4BF861EA        LEA ADST,A5
00612A 4DF86202        LEA ADSTE,A6
                 TRP14 OUT1CR
00612E 1E3C00E3        MOVE.B #OUT1CR,D7
006132 4E4E      TRAP #14
                *
006134 21FC0000621C
    0070         MOVE.L #ADCISR,ADEXADD
                 STPOIN3.W ECGSTRT,A0,PCGSTRT,A1,CARSTRT,A2
00613C 307C0900        MOVE.W #ECGSTRT,A0
006140 327C2100        MOVE.W #PCGSTRT,A1
006144 347C1500        MOVE.W #CARSTRT,A2
                *
                *
                *   INITIALIZE TIMER. THE COUNTER IS SET WITH THE SAMPLING PERI
                *   AND IS DECREMENTED. WHEN THE COUNTER REACHES ZERO, THE ZERO
                *   DETECT BIT IN THE TSR WILL BE SET. THIS INDICATES THE END O
                *   A SAMPLING PERIOD.
                *
                *
                 TIMER A3,A4,ADSMPL,D0
                 STPOIN2.L CNTR,A3,CPR,A4
006148 267C0001002D  MOVE.L #CNTR,A3
00614E 287C00010025  MOVE.L #CPR,A4
006154 707A      MOVE.L #ADSMPL,D0
006156 01CC0000        MOVEP.L D0,0(A4)
00615A 13FC0001
    00010035     MOVE.B #1,TSR
```

```
006162 13FC0001
       00010021  MOVE.B #STRTIMR,TCR
                 *
                 *
                 *  PERFORM A DUMMY WRITE TO INITIATE THE HOLD STATE IN THE
                 *  S/H DEVICES. A DELAY IS THEN INCURRED TO ALLOW THE S/H
                 *  DEVICES TO ACQUIRE  THE SIGNALS.
                 *
                 *
00616A 4202     ADCONT CLR.B D2               CLEAR SIGNAL COUNTER
00616C 13FC0000
       0002000E  MOVE.B #0,HOLD               DUMMY WRITE -- START HOLD STATE
                 DELAY 8,D1                   DELAY FOR SAMPLE-HOLD (37 USEC)
006174 323C0008         MOVE.W #8,D1
006178 51C9FFFE        @010 DBRA D1,@010
                 *
                 *
                 *  DIGITIZE THE ECG, PCG, CAROTID. A DUMMY WRITE IS PERFORMED
                 *  TO TIE THE PROPER MUX INPUT TO THE OUTPUT AND TO START THE
                 *  A/D CONVERTER. A WAIT STATE IS THEN ENTERED UNTIL THE
                 *  PROCESSOR RECEIVES AN INTERRUPT INDICATING THE END OF
                 *  CONVERSION. D2 IS USED AS A SIGNAL COUNTER TO INDICATE WHIC
                 *  SIGNAL IS CURRENTLY BEING DIGITIZED.
                 *
                 *
                 CONVERT ADCH1,D2             CONVERT ECG
00617C 13FC0000
       00020001  MOVE.B #0,ADCH1
006184 4E722300         STOP #$2300
006188 06020001         ADDI.B #1,D2
                 *
                 *
                 CONVERT ADCH2,D2             CONVERT CAROTID
00618C 13FC0000
       00020003  MOVE.B #0,ADCH2
006194 4E722300         STOP #$2300
006198 06020001         ADDI.B #1,D2
                 *
                 *
                 CONVERT ADCH3,D2             CONVERT PCG
00619C 13FC0000
       00020005  MOVE.B #0,ADCH3
0061A4 4E722300         STOP #$2300
0061A8 06020001         ADDI.B #1,D2
                 *
                 *
                 *  WAIT UNTIL THE SAMPLING PERIOD HAS TERMINATED. WHEN THE ZER
                 *  DETECT BIT IN THE TSR EQUALS ONE THE PERIOD IS OVER.
                 *
                 *
0061AC 0C390000
       00010035 ADWAIT CMP.B #$0,TSR
0061B4 67F6      BEQ ADWAIT
                 *
                 *
```

```
                 *
                 *
                 *  AFTER THE PERIOD IS OVER, THE CONTENTS OF THE CPR ARE
                 *  TRANSFERRED TO THE CNTR AND THE CNTR STARTS DECREMENTING
                 *  AGAIN. RESET THE TSR BY WRITING A $1 TO THE REGISTER. THEN
                 *  DETERMINE WHETHER 1.5 SECONDS OF DATA HAS BEEN ACQUIRED. IF
                 *  ALL THE DATA HAS NOT YET BEEN ACQUIRED CONTINUE WITH THIS S
                 *
                 *
0061B6 13FC0001
    00010035  MOVE.B #1,TSR               RESET TSR
0061BE B0FC1500       CMP.W #ECGEND,A0
0061C2 6FA6   BLE ADCONT                  DETERMINE IF ALL DATA ACQUIRED
0061C4 13FC0000
    00010021  MOVE.B #HLTIMR,TCR          HALT TIMER USING TCR HALT CODE
                 *
                 *
                 *  CLEAR MOST SIGNIFICANT NIBBLE IN EACH WORD. THE RANGE OF
                 *  VALUES IS:  -2.5V = 0000; 0.0V = 0800; 2.5V = 0FFF.
                 *
                 *
0061CC 307C0900       MOVE.W #ECGSTRT,A0
0061D0 02580FFF      ADCNVRT ANDI.W #$0FFF,(A0)+  CLEAR MOST SIGNIFICANT NIB
0061D4 B0FC2D00       CMPI.W #PCGEND,A0
0061D8 6FF6   BLE.S ADCNVRT
                 *
              DSPLOPT ADFIN,ADFINE        DISPLAY END OF A/D
0061DA 4BF86204       LEA ADFIN,A5
0061DE 4DF8621A       LEA ADFINE,A6
              TRP14 OUT1CR
0061E2 1E3C00E3       MOVE.B #OUT1CR,D7
0061E6 4E4E   TRAP #14
                 *
0061E8 4E75   RTS
0061EA 53    ADST DC.W 'START OF A/D CONVERSION'
006202 20    ADSTE DC.W ' '
006204 45    ADFIN DC.W 'END OF A/D CONVERSION'
00621A 20    ADFINE DC.W ' '
                 *
                 *
```

```
                *#################################################################
                *                A/D INTERRUPT SERVICE ROUTINE
                *                APU INTERRUPT SERVICE ROUTINE
                *#################################################################
                *
                *     A0 <--> ECG POINTER
                *     A1 <--> PCG POINTER
                *     A2 <--> CAROTID POINTER
                *
                *#################################################################
                *
                *
                *  THIS ISR SERVICES THE INTERRUPT CAUSED BY THE END OF CONVER
                *  SION SIGNAL FROM THE A/D CONVERTER. THE SAMPLE VALUE IS REA
                *  FROM THE A/D BUFFER, THEN STORED IN SYSTEM RAM. THE S/R ALS
                *  KEEPS TRACK OF WHICH SIGNAL IS BEING DIGITIZED AND INCREMEN
                *  THE ECG,PCG,CAROTID POINTERS.  ECG=SIGNAL#0, PCG=SIGNAL#1,
                *  CAROTID=SIGNAL#2.
                *
                *
00621C 0C020000     ADCISR CMPI.B #0,D2
006220 6E08   BGT.S AD1
006222 30F90002000C  MOVE.W ADREAD,(A0)+       STORE ECG DATA AND INCREMENT
006228 6014   BRA.S ADEND
00622A 0C020001     AD1 CMPI.B #1,D2
00622E 6E08   BGT.S AD2
006230 32F90002000C  MOVE.W ADREAD,(A1)+       STORE PCG DATA AND INCREMENT
006236 6006   BRA.S ADEND
006238 34F90002000C AD2 MOVE.W ADREAD,(A2)+    STORE CAROTID DATA & INCREME
00623E 4E73  ADEND RTE
                *
                *  THIS ISR HANDLES THE INTERRUPT RECEIVED FROM THE APU. AN
                *  INTERRUPT ACKNOWLEDGEMENT IS SENT TO THE APU BY WRITING TO
                *  ADDRESS APUEACK.
                *
006240 33FC00FF
       0004000E APISR MOVE.W #$FF,APUEACK  DUMMY WRITE-SEND INTERRUPT ACKNOWLE
006248 4E73     RTE
                *
                *
```

```
                 *##############################################################
                 *##############################################################
                 *
                 *                         ECG  PROCESSING
                 *
                 *##############################################################
                 *
                 *  THE ECG PROCESSING ROUTINE IS COMPRISED OF THE FOLLOWING
                 *  MODULES:
                 *
                 *  EMOVE:  DOWNSAMPLE ECG,CAROTID AND MOVE DATA TO BOTTOM OF
                 *          MEMORY.
                 *  EDFSQU: DIFFERENTIATE AND SQUARE ECG SIGNAL
                 *  ECNVL:  CONVOLVE ECG
                 *  EQRS:   FIND CONSECUTIVE QRS PEAKS
                 *
                 *##############################################################
                 *##############################################################
                 *
                 *
                 * A QRS COMPLEX DETECTION ROUTINE BASED ON A SMOOTHED DIFFEREN
                 * OF THE DIGITIZED ECG SIGNAL IS USED TO DETERMINE THE RR
                 * INTERVAL. WITH X(N) REPRESENTING THE DIGITIZED ECG, THE FIRS
                 * DIFFERENCE IS COMPUTED USING
                 *            D(N)=X(N+1)-X(N).
                 * THE FOLLOWING TRANSFORMATION IS THEN APPLIED TO SMOOTH THE
                 * ENERGY OF THE DIFFERENCE SIGNAL
                 *                   M    2
                 *            G(N)= E   D (N-K)W(K)
                 *                  K=1
                 * WHERE W(K) IS A WEIGHTING SEQUENCE WHICH SELECTS A SEGMENT O
                 * D(N) AND M IS THE NUMBER OF SAMPLES IN WINDOW, AND N IS THE
                 * RUNNING INDEX OF THE SIGNAL AND ITS TRANSFORM.
                 *
                 *
                 *
00624A 4EB872D4      ECG JSR BLANK
                  DSPLOPT ECST,ECEN
00624E 4BF8626E          LEA ECST,A5
006252 4DF86294          LEA ECEN,A6
                  TRP14 OUT1CR
006256 1E3C00E3          MOVE.B #OUT1CR,D7
00625A 4E4E       TRAP #14
00625C 4EB86296          JSR EMOVE
006260 4EB862D4          JSR EDFSQU
006264 4EB862F2          JSR ECNVL
006268 4EB86314          JSR EQRS
00626C 4E75       RTS
00626E 45        ECST DC.W 'ECG PROCESSING - QRS COMPLEX DETECTION'
006294 20        ECEN DC.W ' '
                 *
                 *
```

```
                 *****************************************************************
                 *
                 *                          EMOVE
                 *
                 *****************************************************************
                 *
                 *     A0 <--> SIGNAL POINTER
                 *     A1 <--> END OF SIGNAL DATA
                 *     A2 <--> NEW SIGNAL POINTER
                 *
                 *****************************************************************
                 *
                 *     CALL FOLLOWING MACROS:
                 *          DATAMV->STPOIN2 '''
                 *
                 *****************************************************************
                 *
                 *
                 *  THIS MODULE DOWNSAMPLES THE ECG AND CAROTID BY 4 TO GIVE AN
                 *  EFFECTIVE SAMPLING RATE OF 256 HZ. ALL DATA INCLUDING THE P
                 *  ARE THEN RELOCATED TO THE BOTTOM OF USER MEMORY.
                 *
                 *
                 EMOVE DATAMV ECGSTRT,A0,ECGEND,A1,ECGSNEW,ECGENEW,A2,ECGDWN
                  STPOIN2 ECGSTRT,A0,ECGEND,A1
006296 307C0900        MOVE #ECGSTRT,A0
00629A 327C1500        MOVE #ECGEND,A1
00000900         ECGSTRT SET ECGSNEW
00000C00         ECGEND SET ECGENEW
00629E 347C0900        MOVE.W #ECGSTRT,A2
0062A2 34D0      @016 MOVE.W (A0),(A2)+         MOVE DATA
0062A4 5048       ADD.W #ECGDWN,A0
0062A6 B0C9       CMP.W A1,A0
0062A8 6DF8       BLT.S @016
                  DATAMV CARSTRT,A0,CAREND,A1,CARSNEW,CARENEW,A2,CARDWN
                  STPOIN2 CARSTRT,A0,CAREND,A1
0062AA 307C1500        MOVE #CARSTRT,A0
0062AE 327C2100        MOVE #CAREND,A1
00000C00         CARSTRT SET CARSNEW
00000F00         CAREND SET CARENEW
0062B2 347C0C00        MOVE.W #CARSTRT,A2
0062B6 34D0      @017 MOVE.W (A0),(A2)+         MOVE DATA
0062B8 5048       ADD.W #CARDWN,A0
0062BA B0C9       CMP.W A1,A0
0062BC 6DF8       BLT.S @017
                  DATAMV PCGSTRT,A0,PCGEND,A1,PCGSNEW,PCGENEW,A2,PCGDWN
                  STPOIN2 PCGSTRT,A0,PCGEND,A1
0062BE 307C2100        MOVE #PCGSTRT,A0
0062C2 327C2D00        MOVE #PCGEND,A1
00000F00         PCGSTRT SET PCGSNEW
00001B00         PCGEND SET PCGENEW
0062C6 347C0F00        MOVE.W #PCGSTRT,A2
0062CA 34D0      @018 MOVE.W (A0),(A2)+         MOVE DATA
0062CC 5448       ADD.W #PCGDWN,A0
0062CE B0C9       CMP.W A1,A0
0062D0 6DF8       BLT.S @018
0062D2 4E75       RTS
                 *
                 *
```

```
                   ****************************************************************
                   *                             EDFSQU
                   ****************************************************************
                   *       A0 <--> ECG POINTER
                   *       A1 <--> TRANSFORMED ECG POINTER
                   *       D0 <--> N
                   *       D1 <--> X(N+1)-X(N)
                   *       D2 <--> X(N+1)-X(N)**2
                   ****************************************************************
                   *     CALLS FOLLOWING MACROS:
                   *           STPOIN2         SQUARE
                   ****************************************************************
                   *
                   *  TAKE FIRST DIFFERENCE OF DIGITIZED ECG SIGNAL AND SQUARE.
                   *
                   EDFSQU STPOIN2 ECGSTRT,A0,TRECGST,A1
0062D4 307C0900           MOVE #ECGSTRT,A0
0062D8 327C1B00           MOVE #TRECGST,A1
0062DC 4299         CLR.L (A1)+                    SET 1ST TRANSFORMED ECG SAMPLE TO 0
0062DE 3018        EDFSQ MOVE.W (A0)+,D0
0062E0 3210         MOVE.W (A0),D1
0062E2 9240         SUB.W D0,D1
                    SQUARE D1,D2,A1
0062E4 3401         MOVE.W D1,D2
0062E6 C5C1         MULS.W D1,D2
0062E8 22C2         MOVE.L D2,(A1)+                  MOVE DATA
0062EA B0FC0C00           CMP.W #ECGEND,A0
0062EE 6DEE         BLT.S EDFSQ                    CONTINUE TIL ALL SAMPLES TRANSFORM
0062F0 4E75         RTS
                   *
                   ****************************************************************
                   *                             ECNVL
                   ****************************************************************
                   *     A0 <--> TRANSFORMED ECG START
                   *     A1 <--> TRANSFORMED ECG END
                   *     D0 <--> CONVOLVE WINDOW WIDTH
                   *     A1 <--> TEMPORARY STORGE POINTER
                   ****************************************************************
                   *     CALLS FOLLOWING MACROS:                        SUBROUTINES:
                   *           CALLCNVL        TEMPMOVE                        CNVLV
                   ****************************************************************
                   *
                   *  CALL CONVOLVE ROUTINE TO SMOOTH TRANSFORMED ECG
                   *
                   ECNVL  CALLCNVL TRECGST,ECGCNV,TRECGEN
0062F2 307C1B00           MOVE.W #TRECGST,A0
0062F6 303C0010           MOVE.W #ECGCNV,D0
0062FA 327C20FC           MOVE.W #TRECGEN,A1
0062FE 4EB8608E           JSR CNVLV
                    TEMPMOVE TRECGST,TRECGEN
                    STPOIN2 TRECGST,A0,TMPSTA,A1
006302 307C1B00           MOVE #TRECGST,A0
006306 327C3700           MOVE #TMPSTA,A1
00630A 20D9         @022 MOVE.L (A1)+,(A0)+          MOVE DATA
00630C B0FC20FC           CMP.W #TRECGEN,A0
006310 6DF8         BLT @022
006312 4E75         RTS
                   *
                   *
```

```
                 **************************************************************
                 *                               EQRS
                 **************************************************************
                 *      A0 <--> TRANSFORMED ECG POINTER
                 *      A1 <--> TRANSFORMED ECG SUBPOINTER FOR PEAK DETECT LOOP
                 *      D0 <--> TEMPORARY DATA STORAGE
                 *      D1 <--> MAXIMUM TRANSFORMED ECG VALUE, THEN THRESHOLD
                 *      D2 <--> COUNTER FOR PEAK DETECT LOOP
                 *      D3 <--> NUMBER OF PEAKS DETECTED COUNTER
                 *      D4 <--> PEAK LOCATION COUNTER
                 *      D5 <--> CURRENT TESTING FOR PEAK VALUE
                 *      D6 <--> NUMBER OF SAMPLES TO QRS PEAK #1
                 *      D7 <--> NUMBER OF SAMPLES TO QRS PEAK #2
                 **************************************************************
                 *  FIRST SET POINTERS. THE PEAK LOCATION COUNTER IS FIRST SET
                 *  36 (BYTES). THIS IS DONE SINCE THE SEARCH FOR THE FIRST QRS
                 *  PEAK STARTS AT THE TENTH TRANSFORM SAMPLE. THE QRS PEAK
                 *  COUNTER IS SET TO ONE. ONCE THE COUNTER REACHES -1 BOTH QRS
                 *  COMPLEXES HAVE BEEN FOUND.
                 *
006314 4281      EQRS CLR.L D1                  CLEAR CURRENT MAXIMUM
006316 383C0024          MOVE.W #36,D4          COUNTER
00631A 163C0001          MOVE.B #1,D3           SET QRS PEAK DOWN COUNTER
00631E 307C1B00          MOVE.W #TRECGST,A0 STARTING ADDRESS OF SEARCH FOR MAXI
                 *
                 * THE TRANSFORMED ECG VALUES ARE SCANNED AND A THRESHOLD IS SE
                 * AT THREE QUARTERS THE MAXIMUM VALUE.
                 *
006322 B290      EQRS1 CMP.L (A0),D1            COMPARE CURRENT MAXIMUM TO NEW VAL
006324 6C02       BGE.S EQRS2                   IF (A0)<D1, NO MAXIMUM
006326 2210       MOVE.L (A0),D1                D1<(A1) THUS STORE NEW CURRENT MAX
006328 5848      EQRS2 ADDQ.W #4,A0
000020F0         TRECGCHK EQU TRECGEN-12
00632A B0FC20F0          CMP.W #TRECGCHK,A0
00632E 6DF2       BLT.S EQRS1                   SEARCH ENTIRE DATA
006330 E489       LSR.L #2,D1
006332 2001       MOVE.L D1,D0
006334 D081       ADD.L D1,D0
006336 D081       ADD.L D1,D0                   SET THRESHOLD AT 3/4 * MAX
                 *
                 *
                 * THE TRANSFORMED VALUES ARE SCANNED A SECOND TIME. IF THE SAM
                 * VALUE IS BELOW THE THRESHOLD, THE SAMPLE IS THROWN OUT. NOTE
                 * IF THE ENTIRE TRANSFORM HAS BEEN SCANNED WITHOUT THE DETECTI
                 * OF TWO QRS COMPLEXES, AN ERROR MESSAGE IS DISPLAYED. IF THE
                 * SAMPLE VALUE IS GREATER THAN THE THRESHOLD, THEN CHECK FOR A
                 * PEAK, CORRESPONDING TO QRS COMPLEX.
                 *
                 *
006338 307C1B00          MOVE.W #TRECGST,A0 STARTING ADDRESS OF TRANSFORMED ECG
00633C D0FC0024          ADD.W #36,A0       START LOOKING FOR PEAKS AT 10TH LW
006340 5848      EQRS3 ADDQ.W #4,A0         INCREMENT POINTER TO NEXT LONG WORD
006342 B0FC20FC          CMP.W #TRECGEN,A0
006346 6E46       BGT.S ECGERR              IF ENTIRE TRANSFORMED ECG BUFFER HA
                 *                          BEEN CHECKED W/O 2 PEAKS THEN ERROR
006348 5844       ADDQ.W #4,D4              PEAK LOCATION COUNTER (IN BYTES)
00634A B290       CMP.L (A0),D1             COMPARE (A0) TO THRESHOLD. IF LESS
00634C 6EF2       BGT EQRS3                 THEN GO TO NEXT SAMPLE
                 *
                 *
                 * A PEAK IS FOUND BY COMPARING THE SAMPLE VALUE WITH THE TEN
```

```
                * PREVIOUS AND TEN NEXT SAMPLE VALUES. IF THE SAMPLE VALUE IS
                * GREATER THAN THESE TWENTY SAMPLES, A PEAK, CORRESPONDING TO
                * THE END OF THE QRS COMPLEX, HAS BEEN FOUND. THIS IS CONTINUE
                * UNTIL TWO PEAKS ARE FOUND.
                *
                *
00634E 343C000A       MOVE.W #10,D2        LOAD COUNTER FOR PEAK DETECT LOOP
006352 3248    MOVE.W A0,A1                TRANSFORM ECG POINTER TO SUBPOINTER
006354 2A10    MOVE.L (A0),D5              CURRENT VALUE BEING TESTED FOR PEAK
006356 BAA1   EQRS4 CMP.L -(A1),D5         USING PREDECREMENT
006358 6DE6    BLT EQRS3                   COMPARE VALUE IN (A0) TO 10 PREVIOU
00635A 51CAFFFA       DBRA D2,EQRS4        SAMPLE VALUES.
                *
                *
00635E 343C000A       MOVE.W #10,D2        LOAD COUNTER
006362 3248    MOVE A0,A1                  TRANSFORM ECG POINTER TO SUBPOINTER
006364 BA99   EQRS5 CMP.L (A1)+,D5         AS PREVIOUS ROUTINE. IF VALUE <
006366 6DD8    BLT EQRS3                   10 NEXT SAMPLES, GO TO NEXT SAMPLE.
006368 51CAFFFA       DBRA D2,EQRS5
                *
                *
                * PEAK FOUND!  DETERMINE IF BOTH QRS PEAKS HAVE BEEN FOUND.
                * IF NOT, CONTINUE SEARCH FOR NEXT QRS PEAK.
                *
                *
00636C 3E04    MOVE.W D4,D7                STORE PEAK LOCATION IN D7
00636E 04030001       SUBI.B #1,D3
006372 6D0C    BLT.S EQRS6                 IF =-1, THEN 2 PEAKS HAVE BEEN FOUN
006374 3C07    MOVE.W D7,D6                D6 CONTAINS FIRST PEAK LOCATION
006376 06440190       ADD.W #400,D4
00637A D0FC0190       ADD.W #400,A0        JUMP AHEAD TO SEARCH FOR NEXT PEAK
00637E 60C0    BRA EQRS3                   SEARCH FOR NEXT PEAK
                *
                *
                * BOTH QRS PEAKS HAVE BEEN FOUND. THE ECG HAS BEEN DOWNSAMPLED
                * BY FACTOR OF 4. TAKING INTO ACCOUNT THAT THE TRANSFORM
                * IS REPRESENTED BY LONG WORDS (4 BYTES), THE NUMBER OF BYTES
                * TO QRS1 AND QRS2 CAN BE FOUND BY DIVIDING D6,D7 BY 2.
                *
                *
006380 E24E   EQRS6 LSR.W #1,D6
006382 31C67F00       MOVE.W D6,QRS1       NUMBER OF BYTES TO QRS1 (2 BYTES=
                *                          1 SAMPLE)
006386 E24F    LSR.W #1,D7
006388 31C77F02       MOVE.W D7,QRS2       NUMBER OF BYTES TO QRS2 (2 BYTES=
                *                          1 SAMPLE)
00638C 4E75    RTS
              ECGERR TRP14 ERROR1          ERROR-TRAP 14
00638E 1E3C0000       MOVE.B #ERROR1,D7
006392 4E4E    TRAP #14
                *
                *
```

```
                 ***************************************************************
                 *                    FINDMAX SUBROUTINE
                 *    CALLED BY:  CT1T2
                 ***************************************************************
                 *     A0 <--> SIGNAL POINTER
                 *     A1 <--> END OF SIGNAL
                 *     D0 <--> CURRENT MAXIMUM
                 *     D1 <--> LOCATION OF MAXIMUM IN BYTES
                 *     D2 <--> COUNTER
                 ***************************************************************
                 *     CALLS FOLLOWING MACROS:
                 *           CLDATA3
                 ***************************************************************
                 *
                 *    FINDMAX SUBROUTINE. FINDS MAXIMUM VALUE IN A SET OF DATA
                 *    AND COUNTS THE NUMBER OF BYTES TO THAT MAXIMUM.
                 *
                 FINDMAX CLDATA3.L D0,D1,D2
006394 4280       CLR.L D0
006396 4281       CLR.L D1
006398 4282       CLR.L D2
00639A B090      FINDMX CMP.L (A0),D0       COMPARE TRANSFORM VALUE WITH CURRENT
00639C 6C04       BGE.S FMAX                  IF MAX>VALUE <--> NO NEW MAX
00639E 2010       MOVE.L (A0),D0              IF VALUE>MAX <--> NEW CURRENT MAX
0063A0 3202       MOVE.W D2,D1                LOCATION OF CURRENT MAX IN BYTES
0063A2 5848      FMAX ADDQ.W #4,A0            INCREMENT SIGNAL POINTER
0063A4 5842       ADDQ.W #4,D2                INCREMENT COUNTER
0063A6 B0C9       CMP.W A1,A0                 ALL TRANSFORM VALUES CHECKED?
0063A8 6DF0       BLT FINDMX                 IF NOT CONTINUE
0063AA 4E75       RTS
                 *
                 *
                 ***************************************************************
                 *                     TWO'S COMPLEMENT SUBROUTINE
                 *     CALLED BY: C2DFSQ,PTSQU,PFMSYS
                 ***************************************************************
                 *     A0 <--> STARTING ADDRESS AND SIGNAL POINTER
                 *     A1 <--> ENDING ADDRESS OF SIGNAL
                 *     D0 <--> SAMPLE VALUE TO BE CONVERTED
                 ***************************************************************
                 *
                 *  CONVERTS 12-BIT NUMBER TO 16-BIT TWO'S COMPLEMENT REPRESENT
                 *
0063AC 3010      TWOCOMP MOVE.W (A0),D0
0063AE 06400800          ADDI.W #$800,D0
0063B2 0880000C          BCLR #12,D0
0063B6 0800000B          BTST #11,D0
0063BA 67000006          BEQ TWOCOMA
0063BE 0040F000          OR.W #$F000,D0
0063C2 30C0      TWOCOMA MOVE.W D0,(A0)+       MOVE DATA
0063C4 B0C9       CMP.W A1,A0
0063C6 6FE4       BLE.S TWOCOMP
0063C8 4E75       RTS
                 *
                 *
```

```
                  *###########################################################
                  *###########################################################
                  *
                  *                    CAROTID PROCESSING
                  *
                  *###########################################################
                  *
                  *  THE CAROTID PROCESSING ROUTINE IS COMPRISED OF THE FOLLOWIN
                  *  MODULES:
                  *
                  *    C2DFSQ:  SECOND DERIVATIVE AND SQUARE CAROTID
                  *    CCNVL:   CONVOLVE CAROTID
                  *    CT1T2:   FIND PEAKS IN CAROTID TRANSFORM CORRESPONDING TO
                  *             ONSET OF EJECTION AND DICROTIC NOTCH
                  *    CDCRTC:  FIND DICROTIC NOTCH IN CAROTID PULSE
                  *
                  *###########################################################
                  *
                  *
                  *  THE DICROTIC NOTCH OF THE CAROTID PULSE IS SEARCHED FOR IN
                  *  REGION STARTING AT QRS1 (P1 ON CAROTID PULSE) TO A POINT 50
                  *  MSEC AFTER P1. THE DICROTIC NOTCH IS DEFINED AS THE POINT O
                  *  MINIMUM PRESSURE OCCURRING IN THE REGION OF THE MAXIMUM SEC
                  *  DERIVATIVE. A LEAST SQUARES ESTIMATE OF THE SECOND DEIVATIV
                  *  IS OBTAINED FROM
                  *    P(N) = 2Y(N-2)-Y(N-1)-2Y(N)-Y(N+1)+2Y(N+2)
                  *  WHERE Y(N) IS THE DIGITIZED CAROTID PULSE AND P(N) IS THE
                  *  SECOND DERIVATIVE EVALUATED AT THE NTH DATA POINT. THE
                  *  FOLLOWING TRANSFORMATION IS THEN USED TO OBTAIN A SMOOTHED
                  *  ENERGY CURVE OF THE 2ND DERIVATIVE
                  *                 M  2
                  *            S(N)= E P (N-K)W(K)
                  *                 K=1
                  *  WHERE W(K) IS A WINDOW WHICH SELECTS A SEGMENT OF P(N), M I
                  *  THE NUMBER OF SAMPLES IN WINDOW AND N IS THE RUNNING INDEX
                  *  THE SIGNAL AND TRANSFORM. THE WINDOW W(K) WAS SELECTED SUCH
                  *  THAT  W(K)=(M-K+1).
                  *
                  *
0063CA 4EB872D4       CAR JSR BLANK
                  DSPLOPT CAST,CAEN
0063CE 4BF863EE        LEA CAST,A5
0063D2 4DF8641C        LEA CAEN,A6
                  TRP14 OUT1CR
0063D6 1E3C00E3        MOVE.B #OUT1CR,D7
0063DA 4E4E       TRAP #14
0063DC 4EB8641E        JSR C2DFSQ
0063E0 4EB86470        JSR CCNVL
0063E4 4EB864A0        JSR CT1T2
0063E8 4EB864CA        JSR CDCRTC
0063EC 4E75       RTS
0063EE 43        CAST DC.W 'CAROTID PROCESSING - DICROTIC NOTCH DETECTION'
00641C 20        CAEN DC.W ' '
                  *
                  *
```

```
                  ****************************************************************
                  *
                  *                         C2DFSQ
                  *
                  ****************************************************************
                  *     A0 <--> TRANSFORMED CAROTID POINTER
                  *     A1 <--> START OF CAROTID PULSE CORRESPONDING TO QRS1
                  *     A2 <--> SUBPOINTER IN CALCULATION OF SECOND DIFFERENCE
                  *     A3 <--> END OF TRANSFORMED CAROTID PULSE
                  *     D1 <--> TEMPORARY DATA REGISTER
                  *     D2 <--> TEMPORARY DATA REGISTER
                  ****************************************************************
                  *     CALL THE FOLLOWING MACROS:           SUBROUTINES:
                  *              GETADD          STPOIN2             TWOCOMP
                  *              SQUARE
                  ****************************************************************
                  *  TAKE SECOND DIFFERENCE OF CAROTID PULSE, SQUARE TRANSFORMED
                  *  VALUES IN A REGION STARTING AT QRS1 AND CONTAINING 500 MSEC
                  *  OF DATA.
                  *
00641E 307C0C00       C2DFSQ MOVE.W #CARSTRT,A0
006422 327C0F00        MOVE.W #CAREND,A1
006426 4EB863AC        JSR TWOCOMP            CONVERT CAROTID TO TWOS COMPLEMENT
                   STPOIN2 TRCARST,A0,TRCAREN,A3
00642A 307C2100        MOVE #TRCARST,A0
00642E 367C2300        MOVE #TRCAREN,A3
                   GETADD QRS1,A1,CARSTRT
006432 32787F00        MOVE.W QRS1,A1
006436 D2FC0C00        ADD.W #CARSTRT,A1
00643A 4298        CLR.L (A0)+                CLEAR REGISTER
00643C 4298        CLR.L (A0)+                FIRST TWO LOCATIONS ARE ZERO
00643E 4293        CLR.L (A3)
006440 42A3        CLR.L -(A3)
006442 42A3        CLR.L -(A3)                LAST TWO LOCATIONS ARE ZERO
006444 3449       C2DFSQ1 MOVE A1,A2          USE A2 AS SUBPOINTER
006446 5449        ADDQ.W #2,A1               INCREMENT CAROTID POINTER
006448 321A        MOVE.W (A2)+,D1
00644A 5C4A        ADDQ.W #6,A2
00644C E349        LSL.W #1,D1                COMPUTE  2 * Y(N-2)
00644E 925A        SUB.W (A2)+,D1             - Y(N-1)
006450 5C4A        ADDQ.W #6,A2
006452 341A        MOVE.W (A2)+,D2
006454 5C4A        ADDQ.W #6,A2
006456 E34A        LSL.W #1,D2
006458 9242        SUB.W D2,D1                -2 * Y(N)
00645A 925A        SUB.W (A2)+,D1             - Y(N+1)
00645C 5C4A        ADDQ.W #6,A2
00645E 3412        MOVE.W (A2),D2
006460 E34A        LSL.W #1,D2
006462 D242        ADD.W D2,D1                + 2 * Y(N+2)
                   SQUARE D1,D2,A0            P(N)**2
006464 3401        MOVE.W D1,D2
006466 C5C1        MULS.W D1,D2
006468 20C2        MOVE.L D2,(A0)+            MOVE DATA
00646A B0CB        CMPA.W A3,A0
00646C 6DD6        BLT C2DFSQ1                CONTINUE UNTIL 500 MSEC OF PULSE
00646E 4E75        RTS
                  *
                  *
```

```
                ************************************************************
                *
                *                      CCNVL
                *
                ************************************************************
                *
                *     A0 <--> TRANSFORMED CAROTID START
                *     A1 <--> TRANSFORMED CAROTID END
                *     D0 <--> CONVOLVE WINDOW WIDTH
                *     A1 <--> TEMPORARY MEMORY STORAGE
                *
                ************************************************************
                *
                *     CALLS FOLLOWING MACROS:
                *           CALLCNVL
                *           TEMPMOVE
                *
                *     CALLS FOLLOWING SUBROUTINES:
                *           CONVOLVE
                *
                ************************************************************
                *
                *
                *  CALL CONVOLVE ROUTINE TO SMOOTH TRANSFORMED CAROTID
                *
                CCNVL CALLCNVL TRCARST,CARCNV,TRCAREN
006470 307C2100       MOVE.W #TRCARST,A0
006474 303C0010       MOVE.W #CARCNV,D0
006478 327C2300       MOVE.W #TRCAREN,A1
00647C 4EB8608E       JSR CNVLV
                *
                *
                 TEMPMOVE TRCARST,TRCAREN
                 STPOIN2 TRCARST,A0,TMPSTA,A1
006480 307C2100       MOVE #TRCARST,A0
006484 327C3700       MOVE #TMPSTA,A1
006488 20D9  @030 MOVE.L (A1)+,(A0)+     MOVE DATA
00648A B0FC2300       CMP.W #TRCAREN,A0
00648E 6DF8      BLT @030
                *
                *  CONVERT CAROTID SAMPLES BACK TO 12-BIT REPRESENTATION FOR
                *  DAC ROUTINES
                *
006490 307C0C00       MOVE.W #CARSTRT,A0
006494 06580800      CCNVLA ADD.W #$800,(A0)+    CONVERT TO 12-BIT
006498 B0FC0F00       CMPI.W #CAREND,A0
00649C 6FF6      BLE.S CCNVLA
00649E 4E75      RTS
                *
                *
```

```
                 *************************************************************
                 *
                 *                            CT1T2
                 *
                 *************************************************************
                 *
                 *      A0 <--> TRANSFORMED CAROTID POINTER
                 *      A1 <--> END OF TRANSFORM
                 *      D0 <--> CURRENT MAXIMUM
                 *      D1 <--> OUPEAK, DNPEAK   (T1 & T2)
                 *      D2 <--> COUNTER
                 *
                 *************************************************************
                 *
                 *      CALLS FOLLOWING MACROS:
                 *            GETADD ''        STPOIN2
                 *
                 *      CALLS FOLLOWING SUBROUTINES:
                 *            FINDMAX ''
                 *
                 *************************************************************
                 *
                 *
                 *   FIND PEAK IN CAROTID TRANSFORM CORRESPONDING TO ONSET
                 *   OF EJECTION AND DICROTIC NOTCH, I.E., T1 & T2.
                 *
                 *
                 CT1T2 STPOIN2 TRCARST,A0,TRCAREN,A1
0064A0 307C2100         MOVE #TRCARST,A0
0064A4 327C2300         MOVE #TRCAREN,A1
0064A8 4EB86394         JSR FINDMAX      FIND MAXIMUM VALUE IN TRANSFORMED CARO
                 *
                 *   THE MAXIMUM IN THE CAROTID TRANSFORM CORRESPONDS TO THE ONS
                 *   OF EJECTION. SET A POINTER TO A POINT 150 MSEC (39 SAMPLES
                 *   150 BYTES) AHEAD OF T1. THEN SEARCH FOR T2 IN THE AREA FROM
                 *   THIS POINT TO THE END OF THE TRANSFORMED CAROTID.
                 *
                 *                           D3 CONTAINS NUMBER OF BYTES TO A POINT
                  GETADD D1,D3,T1TOT2    200 MSEC (T1TOT2) AFTER T1.
0064AC 3601       MOVE.W D1,D3
0064AE 06430098         ADD.W #T1TOT2,D3
                  GETADD D3,A0,TRCARST  POINTER FOR SEARCH OF T1
0064B2 3043       MOVE.W D3,A0
0064B4 D0FC2100         ADD.W #TRCARST,A0
0064B8 4EB86394         JSR FINDMAX
0064BC D243       ADD.W D3,D1             BYTES FROM P1 TO T2
0064BE E249       LSR.W #1,D1             DIVIDE BY 2 TO CONVERT LW TO W
0064C0 D2787F00         ADD.W QRS1,D1
0064C4 31C17F04         MOVE.W D1,T2PEAK        BYTES TO T2 FROM BEGINNING OF C
0064C8 4E75       RTS
                 *
                 *
```

```
                   ****************************************************************
                   *
                   *                            CDCRTC
                   *
                   ****************************************************************
                   *
                   *    A0 <--> CAROTID POINTER STARTING 10 SAMPLES BEFORE T2
                   *    A1 <--> POINT 10 SAMPLES AFTER T2
                   *    D0 <--> CURRENT MINIMUM
                   *    D5 <--> START OF S2
                   *
                   ****************************************************************
                   *
                   *     CALLS FOLLOWING MACROS:
                   *           GETADD ''
                   *
                   ****************************************************************
                   *
                   *
                   *  DEFINE A REGION ON THE CAROTID PULSE CORRESPONDING TO +_ 10
                   *  SAMPLES OF THE T2 PEAK. IN THIS REGION SEARCH FOR A LOCAL
                   *  MINIMUM CORRESPONDING TO THE ACTUAL DICROTIC NOTCH
                   *
                   *
                   CDCRTC GETADD T2PEAK,A0,CARSTRT
0064CA 30787F04            MOVE.W T2PEAK,A0
0064CE D0FC0C00            ADD.W #CARSTRT,A0
0064D2 90FC0014            SUBA.W #20,A0     POINT 10 SAMPLES BEFORE T2 ON CAROTID
                    GETADD T2PEAK,A1,CARSTRT
0064D6 32787F04            MOVE.W T2PEAK,A1
0064DA D2FC0C00            ADD.W #CARSTRT,A1
0064DE D2FC0014            ADD.W #20,A1      POINT 10 SAMPLES AFTER T2 ON CAROTID
                   *
0064E2 303C7FFF            MOVE.W #$7FFF,D0 CURRENT MINIMUM VALUE
0064E6 B050   CDCRTC1 CMP.W (A0),D0      COMPARE PULSE VALUE WITH CURRENT MINI
0064E8 6D000006            BLT CDCRTC2       IF VALUE>MIN,THEN NO CURRENT MINIMUM
0064EC 3010    MOVE.W (A0),D0                IF MIN>VALUE,THEN A NEW MINIMUM
0064EE 3A08    MOVE.W A0,D5                  LOCATION OF CURRENT MINIMUM
0064F0 5448   CDCRTC2 ADDQ.W #2,A0           INCREMENT PULSE POINTER
0064F2 B0C9    CMP.W A1,A0                   CHECK ALL 10 SAMPLES FOR MINIMUM
0064F4 6DF0    BLT CDCRTC1
0064F6 04450C00            SUB.W #CARSTRT,D5        SUBTRACT CARST TO GET NUMBER O
                   *                         DICROTIC NOTCH IN CAROTID PULSE
0064FA 31C57F06            MOVE.W D5,DCRTC
0064FE 0445001B            SUB.W #DS2DEL,D5 SUBTRACT DELAY TO GET START OF S2
006502 9A787F00            SUB.W QRS1,D5     CALCULATE NUMBER OF BYTES BEWTEEN QRS
                   *                         AND START OF S2
006506 E54D    LSL.W #2,D5                   MULTIPLY BY 4 (PCG SAMPLED AT RATE 4
                   *                         CAROTID
006508 31C57F0C            MOVE.W D5,PCGS2   NUMBER OF BYTES BETWEEN S1 AND S2
00650C 4E75    RTS
                   *
                   *
```

```
                   **************************************************************
                   *        ENERGY DISTRIBUTION COEFFIECIENT COMPUTATION S/R
                   **************************************************************
                   *      A0 <--> START OF SEGMENT
                   *      A1 <--> END OF SEGMENT
                   *      D3,D2 <--> EDC DENOMINATOR
                   *      D1,D0 <--> EDC NUMERATOR
                   *      D5,D6,D7 <--> TEMP DATA REGISTERS FOR MULTIPLICATION
                   **************************************************************
                   *      CALLS FOLLOWING MACROS:                     SUBROUTINES:
                   *            CLDATA8         ML3216                    DVD6464
                   **************************************************************
                   *
                   * COMPUTE ENERGY DISTRIBUTION COEFFICIENT. PERFORM PRECORRECTI
                   * ON EDC BY MULTIPLYING BY 1024.
                   *
                   EDC CLDATA8.L
 00650E 4280        CLR.L D0
 006510 4281        CLR.L D1
 006512 4282        CLR.L D2
 006514 4283        CLR.L D3
 006516 4284        CLR.L D4
 006518 4285        CLR.L D5
 00651A 4286        CLR.L D6
 00651C 4287        CLR.L D7
 00651E 42B87F12           CLR.L SEQNUM
 006522 2818       EDC1 MOVE.L (A0)+,D4
 006524 D484        ADD.L D4,D2
 006526 D787        ADDX.L D7,D3                  EDC DENOMINATOR
                    ML3216 SEQNUM,D4,D5,D6,D7    E(N)*H(N)
 006528 4286        CLR.L D6
 00652A 3A04        MOVE.W D4,D5
 00652C CAF87F12           MULU.W SEQNUM,D5
 006530 4844        SWAP D4
 006532 C8F87F12           MULU.W SEQNUM,D4
 006536 1E3C000F           MOVE.B #15,D7
 00653A E38C        @037 LSL.L #1,D4
 00653C E396        ROXL.L #1,D6
 00653E 51CFFFFA           DBRA D7,@037
 006542 D885        ADD.L D5,D4
 006544 4287        CLR.L D7
 006546 DD87        ADDX.L D7,D6
 006548 D084        ADD.L D4,D0
 00654A D386        ADDX.L D6,D1.                 EDC NUMERATOR
 00654C 067800017F12  ADDI.W #1,SEQNUM           INCREMENT SEQUENCE
 006552 B0C9        CMPA.W A1,A0
 006554 6DCC        BLT EDC1
 006556 4EB86564           JSR DVD6464
 00655A 20387F44           MOVE.L QUOT,D0
 00655E EB88        LSL.L #5,D0
 006560 EB88        LSL.L #5,D0                   MULTIPLY BY 1024 (PRE-EDC CORRECTI
 006562 4E75        RTS
                   *
                   *
```

```
                  ***************************************************************
                  *                              DVD6464
                  ***************************************************************
                  *     D1,D0 <--> DIVIDEND
                  *     D3,D2 <--> DIVISOR
                  *     D5,D4;D7,D6 <--> PARTIAL DIVIDEND; TEMP PART DIVIDEND
                  ***************************************************************
                  * THIS ROUTINE PERFORMS 16 BIT BY 16 BIT DIVISION. RESULT IS 3
                  * BIT QUOTIENT.
                  *
006564 42B87F44       DVD6464 CLR.L QUOT  CLEAR QUOTIENT
006568 4284    CLR.L D4
00656A 4285    CLR.L D5                   CLEAR PARTIAL DIVIDEND
00656C 11FC00407F48  MOVE.B #64,COUNTER NUMBER OF BYTES IN DIVIDEND
006572 E388   D6464A LSL.L #1,D0
006574 E391    ROXL.L #1,D1
006576 E394    ROXL.L #1,D4
006578 E395    ROXL.L #1,D5               SHIFT DIVIDEND INTO PARTIAL DIVIDEN
00657A 2E387F44       MOVE.L QUOT,D7
00657E E38F    LSL.L #1,D7                SHIFT QUOTIENT
006580 21C77F44       MOVE.L D7,QUOT
006584 2C04    MOVE.L D4,D6
006586 2E05    MOVE.L D5,D7               MOVE PARTIAL DIVIDEND FOR SUBTRACTI
006588 9C82    SUB.L D2,D6
00658A 9F83    SUBX.L D3,D7
00658C 6D00000A       BLT D6464B
006590 2806    MOVE.L D6,D4
006592 2A07    MOVE.L D7,D5               SAVE NEW PARTIAL DIVIDEND
006594 52B87F44       ADDQ.L #1,QUOT      INCREMENT QUOTIENT
006598 53387F48      D6464B SUBQ.B #1,COUNTER   DECREMENT COUNTER
00659C 66D4    BNE D6464A
00659E 4E75    RTS
                  *
                  *
```

```
                *###############################################################
                *###############################################################
                *
                *             PCG PROCESSING - TIME DOMAIN
                *
                *###############################################################
                *
                *  THE PCG PROCESSING ROUTINE IS COMPRISED OF THE FOLLOWING
                *  MODULES:
                *
                *  PTSQU:  SQUARE PCG SAMPLES TO OBTAIN PCG ENERGY CURVE
                *  PTCNVS: SMOOTH SYSTOLIC PCG ENERGY CURVE
                *  PTCNVD: SMOOTH DIASTOLIC PCG ENERGY CURVE
                *  PTEDC:  COMPUTE PCG SYSTOLIC AND DIASTOLIC EDC
                *
                *###############################################################
                *###############################################################
                *
                *
0065A0 4EB872D4      PCGT JSR BLANK
                 DSPLOPT PCST,PCEN
0065A4 4BF865EE        LEA PCST,A5
0065A8 4DF8660A        LEA PCEN,A6
                 TRP14 OUT1CR
0065AC 1E3C00E3        MOVE.B #OUT1CR,D7
0065B0 4E4E      TRAP #14
0065B2 4EB86660        JSR PTSQU
                 DSPLOPT PCSST,PCSEN
0065B6 4BF8660C        LEA PCSST,A5
0065BA 4DF86626        LEA PCSEN,A6
                 TRP14 OUT1CR
0065BE 1E3C00E3        MOVE.B #OUT1CR,D7
0065C2 4E4E      TRAP #14
0065C4 4EB866AE        JSR PTCNVS
                 DSPLOPT PCDST,PCDEN
0065C8 4BF86628        LEA PCDST,A5
0065CC 4DF86644        LEA PCDEN,A6
                 TRP14 OUT1CR
0065D0 1E3C00E3        MOVE.B #OUT1CR,D7
0065D4 4E4E      TRAP #14
0065D6 4EB866DC        JSR PTCNVD
                 DSPLOPT PCEST,PCEEN
0065DA 4BF86646        LEA PCEST,A5
0065DE 4DF8665E        LEA PCEEN,A6
                 TRP14 OUT1CR
0065E2 1E3C00E3        MOVE.B #OUT1CR,D7
0065E6 4E4E      TRAP #14
0065E8 4EB866FE        JSR PTEDC
0065EC 4E75      RTS
0065EE 50       PCST DC.W 'PCG PROCESSING - TIME DOMAIN'
00660A 20       PCEN DC.W ' '
00660C 20       PCSST DC.W '      SYSTOLIC ENERGY CURVE'
006626 20       PCSEN DC.W ' '
006628 20       PCDST DC.W '      DIASTOLIC ENERGY CURVE'
006644 20       PCDEN DC.W ' '
006646 20       PCEST DC.W '      COMPUTE EDC VALUES'
00665E 20       PCEEN DC.W ' '
                *
                *
```

```
                 ***************************************************************
                 *                                PTSQU
                 ***************************************************************
                 *      A0 <--> PCG POINTER STARTING AT QRS1
                 *      A2 <--> PCG ENERGY CURVE POINTER
                 *      A1 <--> PCG END CORRESPONDING TO QRS2
                 *      D0 <--> X(N)
                 *      D1 <--> X(N)**2
                 ***************************************************************
                 *      CALLS FOLLOWING MACROS:
                 *            GETADD ''''       SQUARE
                 ***************************************************************
                 *
                 *   SQUARE ENTIRE PCG SIGNAL TO GET ENERGY CURVE OF SIGNAL. USE
                 *   TWO'S COMPLEMENT REPRESENTATION. OBTAIN START AND END OF ON
                 *   CYCLE OF PCCG CORRESPONDING TO QRS1 AND QRS2.
                 *
                 *
006660 30387F00      PTSQU MOVE.W QRS1,D0
006664 E548      LSL.W #2,D0
006666 31C07F08         MOVE.W D0,PCGS1A
00666A 30387F02         MOVE.W QRS2,D0
00666E E548      LSL.W #2,D0
006670 31C07F0A         MOVE.W D0,PCGS1B     FIND START AND END OF PCG CYCLE
                 *                           CORRESPNDING TO QRS1 AND QRS2. TAKE
                 *                           INTO ACCOUNT PCG SAMPLING RATE.
                  GETADD PCGS1A,A0,PCGSTRT
006674 30787F08         MOVE.W PCGS1A,A0
006678 D0FC0F00         ADD.W #PCGSTRT,A0
                  GETADD PCGS1B,A1,PCGSTRT
00667C 32787F0A         MOVE.W PCGS1B,A1
006680 D2FC0F00         ADD.W #PCGSTRT,A1
006684 4EB863AC         JSR TWOCOMP          CONVERT PCG TO TWOS COMPLEMENT
                  GETADD PCGS1A,A0,PCGSTRT
006688 30787F08         MOVE.W PCGS1A,A0
00668C D0FC0F00         ADD.W #PCGSTRT,A0
                  GETADD PCGS1B,A1,PCGSTRT
006690 32787F0A         MOVE.W PCGS1B,A1
006694 D2FC0F00         ADD.W #PCGSTRT,A1
006698 347C2300         MOVE.W #PCGENST,A2 PCG ENERGY CURVE POINTER
00669C 3018      PTSQ1 MOVE.W (A0)+,D0       GET PCG SAMPLE
                  SQUARE D0,D1,A2
00669E 3200       MOVE.W D0,D1
0066A0 C3C0       MULS.W D0,D1
0066A2 24C1       MOVE.L D1,(A2)+            MOVE DATA
0066A4 B0C9       CMP.W A1,A0                CHECK IF ALL VALUES SQUARED
0066A6 6DF4       BLT PTSQ1                  IF NOT, CONTINUE WITH THIS ROUTINE
0066A8 31CA7F10         MOVE A2,PCGENDE      END OF DIASTOLIC ENERGY CURVE
0066AC 4E75       RTS
                 *
                 *
```

```
                  ***************************************************************
                  *                            PTCNVS
                  ***************************************************************
                  *       A0 <--> START OF SQUARED PCG SIGNAL
                  *       A1 <--> END OF SYSTOLIC SEGMENT
                  *       D0 <--> CONVOLVE WINDOW WIDTH
                  *       A1 <--> TEMPORARY STORAGE POINTER
                  ***************************************************************
                  *     CALLS FOLLOWING MACROS:
                  *           GETADD          STPOIN2 ''
                  *     CALLS FOLLOWING SUBROUTINES:
                  *           CNVLV
                  ***************************************************************
                  *  CALL CONVOLVE ROUTINE TO SMOOTH ENERGY CURVE OF SYSTOLIC
                  *  SEGMENT OF PCG SIGNAL
                  *
                  PTCNVS STPOIN2 PCGENST,A0,PCGCNV,D0
0066AE 307C2300          MOVE #PCGENST,A0
0066B2 303C0020          MOVE #PCGCNV,D0
0066B6 32387F0C          MOVE.W PCGS2,D1
0066BA E349        LSL.W #1,D1
                   GETADD D1,A1,PCGENST
0066BC 3241        MOVE.W D1,A1
0066BE D2FC2300          ADD.W #PCGENST,A1
0066C2 31C97F0E          MOVE A1,PCGENSE  END OF SYSTOLIC ENERGY CURVE
0066C6 4EB8608E          JSR CNVLV        GOTO CONVOLVE SUBROUTINE
                   STPOIN2 PCGENST,A0,TMPSTA,A1
0066CA 307C2300          MOVE #PCGENST,A0
0066CE 327C3700          MOVE #TMPSTA,A1
0066D2 20D9       PTCS1 MOVE.L (A1)+,(A0)+   MOVE CONVOLVE TEMP TO PCG ENERGY
0066D4 B0F87F0E          CMP.W PCGENSE,A0
0066D8 6DF8        BLT PTCS1
0066DA 4E75        RTS
                  *
                  ***************************************************************
                  *                            PTCNVD
                  ***************************************************************
                  *       A0 <--> START OF DIASTOIC SEGMENT OF PCG SIGNAL
                  *       A1 <--> END OF DIASTOLIC SEGMENT
                  *       D0 <--> CONVOLVE WINDOW WIDTH
                  *       A1 <--> TEMPORARY STORAGE POINTER
                  ***************************************************************
                  *     CALLS FOLLOWING SUBROUTINES:
                  *           CNVLV
                  ***************************************************************
                  *  CALL CONVOLVE ROUTINE TO SMOOTH ENERGY CURVE OF DIASTOLIC
                  *  SEGMENT OF PCG SIGNAL
                  *
0066DC 30787F0E         PTCNVD MOVE.W PCGENSE,A0
0066E0 303C0020          MOVE.W #PCGCNV,D0      CONVOLVE WINDOW WIDTH
0066E4 32787F10          MOVE.W PCGENDE,A1      END OF DIASTOLIC SEGMENT
0066E8 4EB8608E          JSR CNVLV       GOTO CONVOLVE SUBROUTINE
0066EC 30787F0E          MOVE.W PCGENSE,A0
0066F0 327C3700          MOVE.W #TMPSTA,A1
0066F4 20D9       PTCD1 MOVE.L (A1)+,(A0)+  MOVE CONVOLVE TEMP TO PCG ENERGY
0066F6 B0F87F10          CMP.W PCGENDE,A0
0066FA 6DF8        BLT PTCD1
0066FC 4E75        RTS
                  *
                  *
```

```
                    ****************************************************************
                    *                       PTEDC
                    ****************************************************************
                    *      A0 <--> START OF SEGMENT
                    *      A1 <--> END OF SEGMENT
                    *      D0 <--> EDC - RETURNED FROM EDC S/R
                    *      D2 <--> NUMBER OF SAMPLES IN SEGMENT
                    ****************************************************************
                    *      CALLS FOLLOWING SUBROUTINES:
                    *            EDC
                    ****************************************************************
                    *
                    *   COMPUTE EDCTSYS
                    *
0066FE 307C2300          PTEDC MOVE.W #PCGENST,A0
006702 32787F0E            MOVE.W PCGENSE,A1
006706 4EB8650E            JSR EDC
                    *
                    * SYSTOLIC EDC IN D0. CORRECT THE EDC BY DIVIDING BY THE NUMBE
                    * OF SAMPLES IN THE SYSTOLIC SEGMENT.
                    *
00670A 323C2300            MOVE.W #PCGENST,D1
00670E 34387F0E            MOVE.W PCGENSE,D2
006712 9441     SUB.W D1,D2
006714 E44A     LSR.W #2,D2
006716 80C2     DIVU D2,D0
006718 31C07F16            MOVE.W D0,EDCTSYS
                    *
                    * COMPUTE EDCTDIA
                    *
00671C 30787F0E            MOVE.W PCGENSE,A0
006720 32787F10            MOVE.W PCGENDE,A1
006724 4EB8650E            JSR EDC
                    *
                    * DIASTOLIC EDC IN D0. CORRECT DIASTOLIC EDC BY DIVIDING BY TH
                    * NUMBER OF SAMPLES IN SEGMENT.
                    *
006728 32387F0E            MOVE.W PCGENSE,D1
00672C 34387F10            MOVE.W PCGENDE,D2
006730 9441     SUB.W D1,D2
006732 E44A     LSR.W #2,D2
006734 80C2     DIVU D2,D0
006736 31C07F18            MOVE.W D0,EDCTDIA
                    *
                GETADD PCGS1A,A0,PCGSTRT  CONVERT BACK TO 12-BIT NUMBER
00673A 30787F08            MOVE.W PCGS1A,A0
00673E D0FC0F00            ADD.W #PCGSTRT,A0
                GETADD PCGS1B,A1,PCGSTRT
006742 32787F0A            MOVE.W PCGS1B,A1
006746 D2FC0F00            ADD.W #PCGSTRT,A1
00674A 06580800          PTEDC1 ADD.W #$800,(A0)+     ADD DATA
00674E B0C9     CMP.W A1,A0
006750 6FF8     BLE.S PTEDC1
006752 4E75     RTS
                    *
                    *
```

```
1617                 ***************************************************************
1618                 *                     APU/FFT SUBROUTINES
1619                 ***************************************************************
1620                 *
1621                 *     THE FOLLOWING SUBROUTINES ARE USED BY FFT:
1622                 *
1623                 *       CALCW:   COMPUTE W=CMPLX(COS(PI/LE1),-SIN(PI/LE1))
1624                 *       CALCT:   COMPUTE T=F(IP)*U
1625                 *       CALCFIM: COMPUTE IMAGINARY F(IP)=F(I)-T;F(I)=F(I)+T
1626                 *       CALCFRL: COMPUTE REAL F(IP)=F(I)-T;F(I)=F(I)+T
1627                 *       CALCU:   COMPUTE U=U*W
1628                 *
1629                 ***************************************************************
1630                 *
1631                 *     CALLS FOLLOWING MACROS:
1632                 *         APU   APURD   APUWR    APUWR16
1633                 *
1634                 ***************************************************************
1635                 *
1636                 * COMPUTE W=CMPLX(COS(PI/LE1),-SIN(PI/LE1)
1637                 *   REAL: WR=COS(PI/LE1)
1638                 *   IMAG: WI=-SIN(PI/LE1)
1639                 *
1640                 CALCW APU PUPI           PUSH PI ON TO STACK -> TOS
1640 006754 13FC001A
        00050011  MOVE.B #PUPI,APUCOM
1640 00675C 4E722300       STOP #$2300
1641 006760 3C387F2E       MOVE.W LE1,D6      MOVE LE1 TO TOS
1642                  APUWR16 D6,A0
1642 006764 3E3C0001       MOVE.W #$1,D7
1642 006768 1086  @053 MOVE.B D6,(A0)
1642 00676A E09E    ROR.L #$8,D6
1642 00676C 51CFFFFA       DBRA D7,@053
1642 006770 E09E    ROR.L #$8,D6
1642 006772 E09E    ROR.L #$8,D6
1643                  APU FLTS             CONVERT LE1 TO 32-BIT FLOATING POINT
1643 006774 13FC001D
        00050011  MOVE.B #FLTS,APUCOM
1643 00677C 4E722300       STOP #$2300
1644                  APU FDIV             PI/LE1 -> TOS
1644 006780 13FC0013
        00050011  MOVE.B #FDIV,APUCOM
1644 006788 4E722300       STOP #$2300
1645                  APU PTOD             PUSH TOS -> NOS; TOS UNCHANGED
1645 00678C 13FC0037
        00050011  MOVE.B #PTOD,APUCOM
1645 006794 4E722300       STOP #$2300
1646                  APU COS              COS(PI/LE1) -> TOS
1646 006798 13FC0003
        00050011  MOVE.B #COS,APUCOM
1646 0067A0 4E722300       STOP #$2300
1647                  APU XCHD             EXCHANGE TOS AND NOS
1647 0067A4 13FC0039
        00050011  MOVE.B #XCHD,APUCOM
1647 0067AC 4E722300       STOP #$2300
1648                  APU SIN              SIN(PI/LE1) -> TOS
1648 0067B0 13FC0002
        00050011  MOVE.B #SIN,APUCOM
1648 0067B8 4E722300       STOP #$2300
1649                  APU CHSF             -SIN(PI/LE1) -> TOS
1649 0067BC 13FC0015
        00050011  MOVE.B #CHSF,APUCOM
```

```
1649 0067C4 4E722300       STOP #$2300
1650                APURD A0,D6
1650 0067C8 3E3C0003       MOVE.W #$3,D7
1650 0067CC E19E   @061 ROL.L #$8,D6
1650 0067CE 1C10   MOVE.B (A0),D6
1650 0067D0 51CFFFFA       DBRA D7,@061
1651 0067D4 21C67F38       MOVE.L D6,WI
1652                APURD A0,D6
1652 0067D8 3E3C0003       MOVE.W #$3,D7
1652 0067DC E19E   @062 ROL.L #$8,D6
1652 0067DE 1C10   MOVE.B (A0),D6
1652 0067E0 51CFFFFA       DBRA D7,@062
1653 0067E4 21C67F34       MOVE.L D6,WR     STORE WR AND WI 32-BIT FL-P VALUES
1654 0067E8 4E75   RTS
1655               *
1656               * COMPUTE T=F(IP)*U:
1657               *   REAL: TR=(FR(IP)*UR)-(FI(IP)*UI)
1658               *   IMAG: TI=(FI(IP)*UR)+(FR(IP)*UR)
1659               *
1660               CALCT APUSGWR A3,D1,A0    MOVE FI(IP) & FR(IP) TO APU STACK
1660 0067EA 2C331000       MOVE.L 0(A3,D1),D6
1660 0067EE 3E3C0003       MOVE.W #$3,D7
1660 0067F2 1086   @063 MOVE.B D6,(A0)
1660 0067F4 E09E   ROR.L #$8,D6
1660 0067F6 51CFFFFA       DBRA D7,@063
1661                APUSGWR A2,D1,A0
1661 0067FA 2C321000       MOVE.L 0(A2,D1),D6
1661 0067FE 3E3C0003       MOVE.W #$3,D7
1661 006802 1086   @064 MOVE.B D6,(A0)
1661 006804 E09E   ROR.L #$8,D6
1661 006806 51CFFFFA       DBRA D7,@064
1662                APUWR D2,A0             MOVE UR TO APU STACK
1662 00680A 3E3C0003       MOVE.W #$3,D7
1662 00680E 1082   @065 MOVE.B D2,(A0)
1662 006810 E09A   ROR.L #$8,D2
1662 006812 51CFFFFA       DBRA D7,@065
1663                APU FMUL                (FR(IP)*UR) -> TOS
1663 006816 13FC0012
          00050011  MOVE.B #FMUL,APUCOM
1663 00681E 4E722300       STOP #$2300
1664                APU XCHD                EXCHANGE TOS AND NOS
1664 006822 13FC0039
          00050011  MOVE.B #XCHD,APUCOM
1664 00682A 4E722300       STOP #$2300
1665                APUWR D3,A0             MOVE UI TO APU STACK
1665 00682E 3E3C0003       MOVE.W #$3,D7
1665 006832 1083   @068 MOVE.B D3,(A0)
1665 006834 E09B   ROR.L #$8,D3
1665 006836 51CFFFFA       DBRA D7,@068
1666                APU FMUL                (FI(IP)*UI) -> TOS
1666 00683A 13FC0012
          00050011  MOVE.B #FMUL,APUCOM
1666 006842 4E722300       STOP #$2300
1667                APU FSUB                (FR(IP)*UR)-(FI(IP)*UI)=TR -> TOS
1667 006846 13FC0011
          00050011  MOVE.B #FSUB,APUCOM
1667 00684E 4E722300       STOP #$2300
1668                APUSGWR A3,D1,A0        MOVE FI(IP) & FR(IP) TO APU STACK
1668 006852 2C331000       MOVE.L 0(A3,D1),D6
1668 006856 3E3C0003       MOVE.W #$3,D7
1668 00685A 1086   @071 MOVE.B D6,(A0)
```

```
00685C E09E     ROR.L #$8,D6
00685E 51CFFFFA        DBRA D7,@071
                APUSGWR A2,D1,A0
006862 2C321000        MOVE.L 0(A2,D1),D6
006866 3E3C0003        MOVE.W #$3,D7
00686A 1086     @072 MOVE.B D6,(A0)
00686C E09E     ROR.L #$8,D6
00686E 51CFFFFA        DBRA D7,@072
                APUWR D3,A0               MOVE UI TO APU STACK
006872 3E3C0003        MOVE.W #$3,D7
006876 1083     @073 MOVE.B D3,(A0)
006878 E09B     ROR.L #$8,D3
00687A 51CFFFFA        DBRA D7,@073
                APU FMUL                  (FR(IP)*UI) -> TOS
00687E 13FC0012
    00050011    MOVE.B #FMUL,APUCOM
006886 4E722300        STOP #$2300
                APU XCHD                  EXCHANGE TOS AND NOS
00688A 13FC0039
    00050011    MOVE.B #XCHD,APUCOM
006892 4E722300        STOP #$2300
                APUWR D2,A0               MOVE UR TO APU STACK
006896 3E3C0003        MOVE.W #$3,D7
00689A 1082     @076 MOVE.B D2,(A0)
00689C E09A     ROR.L #$8,D2
00689E 51CFFFFA        DBRA D7,@076
                APU FMUL                  (FI(IP)*UR) -> TOS
0068A2 13FC0012
    00050011    MOVE.B #FMUL,APUCOM
0068AA 4E722300        STOP #$2300
                APU FADD                  (FR(IP)*UI)+(FI(IP)*UR)=TI -> TOS
0068AE 13FC0010
    00050011    MOVE.B #FADD,APUCOM
0068B6 4E722300        STOP #$2300
0068BA 4E75     RTS
               *
               * COMPUTE IMAGINARY PART OF F(IP)=F(I)-T;F(I)=F(I)+T:
               *                             FI(IP)=FI(I)-TI;FI(I)=FI(I)+TI
               *
               CALCFIM APU PTOD           PUSH TOS->NOS; TOS UNCHANGED
0068BC 13FC0037
    00050011    MOVE.B #PTOD,APUCOM
0068C4 4E722300        STOP #$2300
                APUSGWR A3,D0,A0          MOVE F(I) TO APU STACK
0068C8 2C330000        MOVE.L 0(A3,D0),D6
0068CC 3E3C0003        MOVE.W #$3,D7
0068D0 1086     @080 MOVE.B D6,(A0)
0068D2 E09E     ROR.L #$8,D6
0068D4 51CFFFFA        DBRA D7,@080
                APU XCHD                  EXCHANGE TOS AND NOS
0068D8 13FC0039
    00050011    MOVE.B #XCHD,APUCOM
0068E0 4E722300        STOP #$2300
                APU FSUB                  FI(I)-TI=FI(IP)
0068E4 13FC0011
    00050011    MOVE.B #FSUB,APUCOM
0068EC 4E722300        STOP #$2300
                APU XCHD                  EXCHANGE TOS AND NOS
0068F0 13FC0039
    00050011    MOVE.B #XCHD,APUCOM
0068F8 4E722300        STOP #$2300
```

```
                     APUSGWR A3,D0,A0          MOVE F(I) TO APU STACK
 0068FC 2C330000          MOVE.L 0(A3,D0),D6
 006900 3E3C0003          MOVE.W #$3,D7
 006904 1086   @084 MOVE.B D6,(A0)
 006906 E09E   ROR.L #$8,D6
 006908 51CFFFFA          DBRA D7,@084
                     APU FADD                  FI(I)+TI=FI(I)
 00690C 13FC0010
        00050011  MOVE.B #FADD,APUCOM
 006914 4E722300          STOP #$2300
                     APUSGRD A3,D0,A0
 006918 3E3C0003          MOVE.W #$3,D7
 00691C E19E   @086 ROL.L #$8,D6
 00691E 1C10   MOVE.B (A0),D6
 006920 51CFFFFA          DBRA D7,@086
 006924 27860000          MOVE.L D6,0(A3,D0)
                     APUSGRD A3,D1,A0
 006928 3E3C0003          MOVE.W #$3,D7
 00692C E19E   @087 ROL.L #$8,D6
 00692E 1C10   MOVE.B (A0),D6
 006930 51CFFFFA          DBRA D7,@087
 006934 27861000          MOVE.L D6,0(A3,D1)
 006938 4E75   RTS
                     *
                     * COMPUTE REAL PART OF F(IP)=F(I)-T;F(I)=F(I)+T:
                     *                         FR(IP)=FR(I)-TR;FR(I)=FR(I)+TR
                     *
                     CALCFRL APU PTOD          PUSH TOS->NOS; TOS UNCHANGED
 00693A 13FC0037
        00050011  MOVE.B #PTOD,APUCOM
 006942 4E722300          STOP #$2300
                     APUSGWR A2,D0,A0          MOVE FR(I) TO APU STACK
 006946 2C320000          MOVE.L 0(A2,D0),D6
 00694A 3E3C0003          MOVE.W #$3,D7
 00694E 1086   @089 MOVE.B D6,(A0)
 006950 E09E   ROR.L #$8,D6
 006952 51CFFFFA          DBRA D7,@089
                     APU XCHD                  EXCHANGE TOS AND NOS
 006956 13FC0039
        00050011  MOVE.B #XCHD,APUCOM
 00695E 4E722300          STOP #$2300
                     APU FSUB                  FR(I)-TR=FR(IP)
 006962 13FC0011
        00050011  MOVE.B #FSUB,APUCOM
 00696A 4E722300          STOP #$2300
                     APU XCHD                  EXCHANGE TOS AND NOS
 00696E 13FC0039
        00050011  MOVE.B #XCHD,APUCOM
 006976 4E722300          STOP #$2300
                     APUSGWR A2,D0,A0          MOVE FR(I) TO APU STACK
 00697A 2C320000          MOVE.L 0(A2,D0),D6
 00697E 3E3C0003          MOVE.W #$3,D7
 006982 1086   @093 MOVE.B D6,(A0)
 006984 E09E   ROR.L #$8,D6
 006986 51CFFFFA          DBRA D7,@093
                     APU FADD                  FR(I)+TR=FR(I)
 00698A 13FC0010
        00050011  MOVE.B #FADD,APUCOM
 006992 4E722300          STOP #$2300
                     APUSGRD A2,D0,A0
 006996 3E3C0003          MOVE.W #$3,D7
```

```
1702 00699A E19E   @095 ROL.L #$8,D6
1702 00699C 1C10    MOVE.B (A0),D6
1702 00699E 51CFFFFA       DBRA D7,@095
1702 0069A2 25860000       MOVE.L D6,0(A2,D0)
1703                APUSGRD A2,D1,A0
1703 0069A6 3E3C0003       MOVE.W #$3,D7
1703 0069AA E19E   @096 ROL.L #$8,D6
1703 0069AC 1C10    MOVE.B (A0),D6
1703 0069AE 51CFFFFA       DBRA D7,@096
1703 0069B2 25861000       MOVE.L D6,0(A2,D1)
1704 0069B6 4E75    RTS
1705               *
1706               * COMPUTE U=U*W
1707               *   REAL: UR=(UR*WR)-(UI*WI)
1708               *   IMAG: UI=(UI*WR)+(UR*WI)
1709               *
1710               CALCU APUWR D2,A0         MOVE UI & WI TO APU STACK
1710 0069B8 3E3C0003       MOVE.W #$3,D7
1710 0069BC 1082   @097 MOVE.B D2,(A0)
1710 0069BE E09A    ROR.L #$8,D2
1710 0069C0 51CFFFFA       DBRA D7,@097
1711 0069C4 2C387F34       MOVE.L WR,D6
1712                APUWR D6,A0
1712 0069C8 3E3C0003       MOVE.W #$3,D7
1712 0069CC 1086   @098 MOVE.B D6,(A0)
1712 0069CE E09E    ROR.L #$8,D6
1712 0069D0 51CFFFFA       DBRA D7,@098
1713                APU FMUL                 (UI*WI) -> TOS
1713 0069D4 13FC0012
        00050011    MOVE.B #FMUL,APUCOM
1713 0069DC 4E722300       STOP #$2300
1714                APUWR D3,A0              MOVE UR & WR TO APU STACK
1714 0069E0 3E3C0003       MOVE.W #$3,D7
1714 0069E4 1083   @100 MOVE.B D3,(A0)
1714 0069E6 E09B    ROR.L #$8,D3
1714 0069E8 51CFFFFA       DBRA D7,@100
1715 0069EC 2C387F38       MOVE.L WI,D6
1716                APUWR D6,A0
1716 0069F0 3E3C0003       MOVE.W #$3,D7
1716 0069F4 1086   @101 MOVE.B D6,(A0)
1716 0069F6 E09E    ROR.L #$8,D6
1716 0069F8 51CFFFFA       DBRA D7,@101
1717                APU FMUL                 (UR*WR) -> TOS
1717 0069FC 13FC0012
        00050011    MOVE.B #FMUL,APUCOM
1717 006A04 4E722300       STOP #$2300
1718                APU FSUB                 (UR*WR)-(UI*WI)=UR->TOS
1718 006A08 13FC0011
        00050011    MOVE.B #FSUB,APUCOM
1718 006A10 4E722300       STOP #$2300
1719                APUWR D3,A0              MOVE UI & WR TO APU STACK
1719 006A14 3E3C0003       MOVE.W #$3,D7
1719 006A18 1083   @104 MOVE.B D3,(A0)
1719 006A1A E09B    ROR.L #$8,D3
1719 006A1C 51CFFFFA       DBRA D7,@104
1720 006A20 2C387F34       MOVE.L WR,D6
1721                APUWR D6,A0
1721 006A24 3E3C0003       MOVE.W #$3,D7
1721 006A28 1086   @105 MOVE.B D6,(A0)
1721 006A2A E09E    ROR.L #$8,D6
1721 006A2C 51CFFFFA       DBRA D7,@105
```

```
1722                  APU FMUL                    (UI*WR) -> TOS
1722 006A30 13FC0012
        00050011  MOVE.B #FMUL,APUCOM
1722 006A38 4E722300       STOP #$2300
1723                  APUWR D2,A0                 MOVE UR & WI TO APU STACK
1723 006A3C 3E3C0003       MOVE.W #$3,D7
1723 006A40 1082   @107 MOVE.B D2,(A0)
1723 006A42 E09A    ROR.L #$8,D2
1723 006A44 51CFFFFA       DBRA D7,@107
1724 006A48 2C387F38       MOVE.L WI,D6
1725                  APUWR D6,A0
1725 006A4C 3E3C0003       MOVE.W #$3,D7
1725 006A50 1086   @108 MOVE.B D6,(A0)
1725 006A52 E09E    ROR.L #$8,D6
1725 006A54 51CFFFFA       DBRA D7,@108
1726                  APU FMUL                    (UR*WI ) -> TOS
1726 006A58 13FC0012
        00050011  MOVE.B #FMUL,APUCOM
1726 006A60 4E722300       STOP #$2300
1727                  APU FADD                    (UI*WR)+(UR*WI )=UI->TOS
1727 006A64 13FC0010
        00050011  MOVE.B #FADD,APUCOM
1727 006A6C 4E722300       STOP #$2300
1728                  APURD A0,D3                 NEW UR & UI VALUES
1728 006A70 3E3C0003       MOVE.W #$3,D7
1728 006A74 E19B   @111 ROL.L #$8,D3
1728 006A76 1610    MOVE.B (A0),D3
1728 006A78 51CFFFFA       DBRA D7,@111
1729                  APURD A0,D2
1729 006A7C 3E3C0003       MOVE.W #$3,D7
1729 006A80 E19A   @112 ROL.L #$8,D2
1729 006A82 1410    MOVE.B (A0),D2
1729 006A84 51CFFFFA       DBRA D7,@112
1730 006A88 4E75    RTS
```

```
1732                 *************************************************************
1733                 *                         FFT SUBROUTINE
1734                 *************************************************************
1735                 *      A0 <--> APU OPERAND ADDRESS
1736                 *      A1 <--> APU COMMAND/STATUS ADDRESS
1737                 *************************************************************
1738                 *      CALLS FOLLOWING MACROS:   STPOIN2
1739                 *************************************************************
1740                 *
1741                 FFT DSPLOPT HAM,HAME
1741 006A8A 4BF86E9C        LEA HAM,A5
1741 006A8E 4DF86EB4        LEA HAME,A6
1741                  TRP14 OUT1CR
1741 006A92 1E3C00E3        MOVE.B #OUT1CR,D7
1741 006A96 4E4E      TRAP #14
1742                  STPOIN2.L APUOPER,A0,APUCOM,A1
1742 006A98 207C00050001  MOVE.L #APUOPER,A0
1742 006A9E 227C00050011  MOVE.L #APUCOM,A1
1743                 *
1744                 *  CONVERT VALUES IN REAL AND IMAGINARY ARRAYS FROM 16-BIT
1745                 *  FIXED POINT TO 32-BIT FLOATING POINT. MULTIPLY SIGNAL
1746                 *  BY HAMMING WINDOW W(N), WHERE W(N)=.54+.46(COS(2*PI*N/M))
1747                 *
1748                  STPOIN2.W XRST,A2,XIST,A3   START OF REAL AND IMAGINARY
1748 006AA4 347C3700        MOVE.W #XRST,A2
1748 006AA8 367C4700        MOVE.W #XIST,A3
1749                 *
1750                 * CONVERT 16-BIT NUMBERS IN REAL AND IMAGINARY ARRAYS
1751                 * TO 32-BIT FLOATING POINT
1752                 *
1753 006AAC 2012     FFT0 MOVE.L (A2),D0        PUT 16-BIT NUMBER INTO DR
1754                  APUWR16 D0,A0             WRITE NUMBER TO STACK
1754 006AAE 3E3C0001        MOVE.W #$1,D7
1754 006AB2 1080     @116 MOVE.B D0,(A0)
1754 006AB4 E098      ROR.L #$8,D0
1754 006AB6 51CFFFFA        DBRA D7,@116
1754 006ABA E098      ROR.L #$8,D0
1754 006ABC E098      ROR.L #$8,D0
1755                  APU FLTS                  CONVERT TO FLOATING POINT
1755 006ABE 13FC001D
       00050011   MOVE.B #FLTS,APUCOM
1755 006AC6 4E722300        STOP #$2300
1756                  APURD A0,D0               READ CONVERTED NUMBER
1756 006ACA 3E3C0003        MOVE.W #$3,D7
1756 006ACE E198     @118 ROL.L #$8,D0
1756 006AD0 1010      MOVE.B (A0),D0
1756 006AD2 51CFFFFA        DBRA D7,@118
1757 006AD6 24C0      MOVE.L D0,(A2)+           STORE BACK INTO ARRAY
1758 006AD8 B4FC5700        CMP.W #XIEN,A2
1759 006ADC 6DCE      BLT FFT0                  CONTINUE UNTIL ENTIRE ARRAYS CONVERTE
1760                 *
1761                 * MULTIPLY BY HAMMING WINDOW
1762                 *
1763 006ADE 2A3C00000400 HAMM MOVE.L #1024,D5       NUMBER OF SAMPLES IN SIGNAL
1764 006AE4 223CFFFFFE01  MOVE.L #-511,D1   START OF HAMMING WINDOW
1765 006AEA 263C008A3D70  MOVE.L #HAM54,D3
1766 006AF0 283C7FEB851E  MOVE.L #HAM46,D4
1767                  APU PUPI
1767 006AF6 13FC001A
       00050011   MOVE.B #PUPI,APUCOM
1767 006AFE 4E722300        STOP #$2300
```

```
1768 006B02 7002    MOVE.L #$2,D0
1769                APUWR D0,A0
1769 006B04 3E3C0003        MOVE.W #$3,D7
1769 006B08 1080  @120 MOVE.B D0,(A0)
1769 006B0A E098    ROR.L #$8,D0
1769 006B0C 51CFFFFA        DBRA D7,@120
1770                APU FLTD
1770 006B10 13FC001C
        00050011  MOVE.B #FLTD,APUCOM
1770 006B18 4E722300        STOP #$2300
1771                APU FMUL
1771 006B1C 13FC0012
        00050011  MOVE.B #FMUL,APUCOM
1771 006B24 4E722300        STOP #$2300
1772                APUWR D5,A0
1772 006B28 3E3C0003        MOVE.W #$3,D7
1772 006B2C 1085  @123 MOVE.B D5,(A0)
1772 006B2E E09D    ROR.L #$8,D5
1772 006B30 51CFFFFA        DBRA D7,@123
1773                APU FLTD
1773 006B34 13FC001C
        00050011  MOVE.B #FLTD,APUCOM
1773 006B3C 4E722300        STOP #$2300
1774                APU FDIV
1774 006B40 13FC0013
        00050011  MOVE.B #FDIV,APUCOM
1774 006B48 4E722300        STOP #$2300
1775                APURD A0,D2                D2 = 2*PI/1024
1775 006B4C 3E3C0003        MOVE.W #$3,D7
1775 006B50 E19A  @126 ROL.L #$8,D2
1775 006B52 1410    MOVE.B (A0),D2
1775 006B54 51CFFFFA        DBRA D7,@126
1776                STPOIN2.W XRST,A2,XREN,A3
1776 006B58 347C3700        MOVE.W #XRST,A2
1776 006B5C 367C4700        MOVE.W #XREN,A3
1777              HAMM1 APUWR D1,A0
1777 006B60 3E3C0003        MOVE.W #$3,D7
1777 006B64 1081  @128 MOVE.B D1,(A0)
1777 006B66 E099    ROR.L #$8,D1
1777 006B68 51CFFFFA        DBRA D7,@128
1778                APU FLTD
1778 006B6C 13FC001C
        00050011  MOVE.B #FLTD,APUCOM
1778 006B74 4E722300        STOP #$2300
1779                APUWR D2,A0
1779 006B78 3E3C0003        MOVE.W #$3,D7
1779 006B7C 1082  @130 MOVE.B D2,(A0)
1779 006B7E E09A    ROR.L #$8,D2
1779 006B80 51CFFFFA        DBRA D7,@130
1780                APU FMUL                   2PI/1024 * N
1780 006B84 13FC0012
        00050011  MOVE.B #FMUL,APUCOM
1780 006B8C 4E722300        STOP #$2300
1781                APU COS                    COS(2PI*N/1024)
1781 006B90 13FC0003
        00050011  MOVE.B #COS,APUCOM
1781 006B98 4E722300        STOP #$2300
1782                APUWR D4,A0
1782 006B9C 3E3C0003        MOVE.W #$3,D7
1782 006BA0 1084  @133 MOVE.B D4,(A0)
1782 006BA2 E09C    ROR.L #$8,D4
```

```
006BA4 51CFFFFA       DBRA D7,@133
              APU FMUL                      .46COS(2PI*N/1024)
006BA8 13FC0012
    00050011  MOVE.B #FMUL,APUCOM
006BB0 4E722300       STOP #$2300
              APUWR D3,A0
006BB4 3E3C0003       MOVE.W #$3,D7
006BB8 1083   @135 MOVE.B D3,(A0)
006BBA E09B   ROR.L #$8,D3
006BBC 51CFFFFA       DBRA D7,@135
              APU FADD                      .54+(.46COS(2PI*N/1024)
006BC0 13FC0010
    00050011  MOVE.B #FADD,APUCOM
006BC8 4E722300       STOP #$2300
006BCC 2012   MOVE.L (A2),D0
              APUWR D0,A0
006BCE 3E3C0003       MOVE.W #$3,D7
006BD2 1080   @137 MOVE.B D0,(A0)
006BD4 E098   ROR.L #$8,D0
006BD6 51CFFFFA       DBRA D7,@137
              APU FMUL                     X(N) * W(N)
006BDA 13FC0012
    00050011  MOVE.B #FMUL,APUCOM
006BE2 4E722300       STOP #$2300
              APURD A0,D0
006BE6 3E3C0003       MOVE.W #$3,D7
006BEA E198   @139 ROL.L #$8,D0
006BEC 1010   MOVE.B (A0),D0
006BEE 51CFFFFA       DBRA D7,@139
006BF2 24C0   MOVE.L D0,(A2)+         MOVE DATA
006BF4 5281   ADD.L #$1,D1
006BF6 B4CB   CMP.W A3,A2
006BF8 6F00FF66       BLE HAMM1
              *
              *****************************************************************
              *      A2 <--> START OF REAL PART OF SIGNAL
              *      A3 <--> START OF IMAGINARY PART OF SIGNAL
              *      D0 <--> J; D1 <--> I; D2 <--> T; D3 <--> K
              *      D4 <--> (J-1)*4 ARRAY POINTER
              *      D5 <--> (I-1)*4 ARRAY POINTER
              *****************************************************************
              *      CALLS FOLLOWING MACROS:
              *            STPOIN2 ''      CLDATA2           REORDER ''
              *****************************************************************
              *
              *     FFT SUBROUTINE  ----  DO 3 LOOP
              *
              DSPLOPT FFTM,FFTME
006BFC 4BF86EB6       LEA FFTM,A5
006C00 4DF86EC4       LEA FFTME,A6
              TRP14 OUT1CR
006C04 1E3C00E3       MOVE.B #OUT1CR,D7
006C08 4E4E   TRAP #14
006C0A 207C00050001   MOVE.L #APUOPER,A0
006C10 347C3700       MOVE.W #XRST,A2  START OF REAL ARRAY
              CLDATA2.W D4,D5
006C14 4244   CLR.W D4
006C16 4245   CLR.W D5
              STPOIN2 1,D0,1,D1        J=1;I=1
006C18 303C0001       MOVE #1,D0
006C1C 323C0001       MOVE #1,D1
```

```
                     *
                     * DO  3  I = 1,NM1
                     *
006C20 B240          FFT3LP CMP.W D0,D1
006C22 6C1C            BGE.S FFT1                IF (I .GE. J) GOTO 1
                       REORDER A2,D4,D2,D5       T=F(J);F(J)=F(I);F(I)=T:*REAL*
006C24 24324000              MOVE.L 0(A2,D4),D2
006C28 25B250004000          MOVE.L 0(A2,D5),0(A2,D4)
006C2E 25825000              MOVE.L D2,0(A2,D5)
                       REORDER A3,D4,D2,D5       :*IMAGINARY*
006C32 24334000              MOVE.L 0(A3,D4),D2
006C36 27B350004000          MOVE.L 0(A3,D5),0(A3,D4)
006C3C 27825000              MOVE.L D2,0(A3,D5)
006C40 363C0200             FFT1 MOVE.W #NV2,D3        K=NV2
006C44 B640          FFT2 CMP.W D0,D3
006C46 6C06            BGE.S FFT3                IF (K .GE. J) GOTO 3
006C48 9043            SUB.W D3,D0               J=J-K
006C4A E24B            LSR.W #1,D3               K=K/2
006C4C 60F6            BRA.S FFT2                GOTO 2
006C4E D043          FFT3 ADD.W D3,D0            J=J+K
006C50 5241            ADDQ.W #1,D1              I=I+1
006C52 5845            ADDQ.W #$4,D5             I ARRAY POINTER
006C54 3800            MOVE.W D0,D4
006C56 5344            SUBQ.W #$1,D4
006C58 E54C            LSL.W #$2,D4              J ARRAY POINTER
006C5A 0C4103FF              CMP.W #NM1,D1
006C5E 6FC0            BLE.S FFT3LP              3 CONTINUE
                     *
                     *
                     *
                     *
                     ****************************************************************
                     *      A0,A1 <--> APU OPERAND ADDRESS, APU COMMAND ADDRESS
                     *      A2,A3 <--> REAL AND IMAGINARY SIGNAL START
                     *      D0,D1 <--> I ARRAY SUBSCRIPT (LW), IP ARRAY SCRIPT (LW)
                     *      D2,D3 <--> UR,UI 32-BIT FLOATING POINT VALUES
                     *      D6,D7 <--> SCRATCH
                     ****************************************************************
                     *
                     *         DO  5  OUTSIDE  LOOP
                     *
                       STPOIN2 XRST,A2,XIST,A3  START OF REAL AND IMAGINARY SIGNAL
006C60 347C3700              MOVE #XRST,A2
006C64 367C4700              MOVE #XIST,A3
                     *
                     * DO 5 L=1,LN
                     *
006C68 31FC00017F2A  MOVE.W #1,L                 L=1
                     FFTOLP5 POW2TOA L
006C6E 3E387F2A              MOVE.W L,D7
006C72 04470001              SUBI.W #1,D7
006C76 3C3C0001              MOVE.W #1,D6
006C7A CCFC0002             @146 MULU.W #2,D6
006C7E 51CFFFFA              DBRA D7,@146
006C82 31C67F2C              MOVE.W D6,LE        LE=2**L
006C86 E24E            LSR.W #1,D6
006C88 31C67F2E              MOVE D6,LE1         LE1=LE/2
006C8C 42B87F3C              CLR.L UR
006C90 06B800000001
    7F3C               ADDI.L #1,UR
006C98 42B87F40              CLR.L UI            U=(1.0,0.0)
```

```
006C9C 2C387F3C       MOVE.L UR,D6       MOVE UR & UI TO APU STACK
                APUWR D6,A0
006CA0 3E3C0003       MOVE.W #$3,D7
006CA4 1086    @147 MOVE.B D6,(A0)
006CA6 E09E     ROR.L #$8,D6
006CA8 51CFFFFA       DBRA D7,@147
006CAC 2C387F40       MOVE.L UI,D6
                APUWR D6,A0
006CB0 3E3C0003       MOVE.W #$3,D7
006CB4 1086    @148 MOVE.B D6,(A0)
006CB6 E09E     ROR.L #$8,D6
006CB8 51CFFFFA       DBRA D7,@148
                APU FLTD                 CONVERT UI TO 32-BIT FLOATING POINT
006CBC 13FC001C
       00050011  MOVE.B #FLTD,APUCOM
006CC4 4E722300       STOP #$2300
                APU XCHD                 EXCHANGE TOS & NOS
006CC8 13FC0039
       00050011  MOVE.B #XCHD,APUCOM
006CD0 4E722300       STOP #$2300
                APU FLTD                 CONVERT UR TO 32-BIT FLOATING POINT
006CD4 13FC001C
       00050011  MOVE.B #FLTD,APUCOM
006CDC 4E722300       STOP #$2300
                APURD A0,D2              UR & UI 32-BIT FL-P VALUES
006CE0 3E3C0003       MOVE.W #$3,D7
006CE4 E19A    @152 ROL.L #$8,D2
006CE6 1410     MOVE.B (A0),D2
006CE8 51CFFFFA       DBRA D7,@152
                APURD A0,D3
006CEC 3E3C0003       MOVE.W #$3,D7
006CF0 E19B    @153 ROL.L #$8,D3
006CF2 1610     MOVE.B (A0),D3
006CF4 51CFFFFA       DBRA D7,@153
               *
               * COMPUTE W=CMPLX(COS(PI/LE1),-SIN(PI/LE1))
               *
006CF8 4EB86754       JSR CALCW
               *
               *      DO 5 J=1,LE1
               *
006CFC 31FC00017F32  MOVE.W #1,J       J=1
               *
               *      DO 4 I=J,N,LE
               *
006D02 31F87F327F30 FFTLP5 MOVE.W J,I I=J
006D08 30387F30      FFTLP4 MOVE.W I,D0
006D0C 04400001       SUBI.W #1,D0
006D10 E548     LSL.W #2,D0              CONVERT TO LW ARRAY POINTER
006D12 32387F30       MOVE.W I,D1
006D16 D2787F2E       ADD.W LE1,D1       IP=I+LE1
006D1A 04410001       SUBI.W #1,D1
006D1E E549     LSL.W #2,D1              CONVERT TO LW ARRAY POINTER
               *
               *   COMPUTE T=F(IP)*U:
               *
006D20 4EB867EA       JSR CALCT
               *
               *   TI -> TOS; TR -> NOS
               *   COMPUTE F(IP)=F(I)-T
               *           F(I)=F(I)+T
```

```
                  *
006D24 4EB868BC        JSR CALCFIM
006D28 4EB8693A        JSR CALCFRL
006D2C 3C387F2C      FFT4 MOVE.W LE,D6
006D30 DD787F30        ADD.W D6,I         I=I+LE
006D34 0C7804007F30  CMP.W #N,I
006D3A 6FCC    BLE FFTLP4                 CONTINUE UNTIL LOOP 4 UNTIL I>N
                  *
                  *   COMPUTE U=U*W:
                  *
006D3C 4EB869B8        JSR CALCU
                  *
006D40 067800017F32 FFT5 ADDI.W #1,J  J=J+1
006D46 3C387F32        MOVE.W J,D6
006D4A BC787F2E        CMP.W LE1,D6
006D4E 6FB2    BLE FFTLP5                 CONTINUE INNER LOOP 5 UNTIL J>LE1
006D50 067800017F2A FFTO5 ADDI.W #1,L
006D56 0C78000A7F2A  CMPI.W #LNN,L
006D5C 6F00FF10        BLE FFTOLP5        CONTINUE OUTER LOOP 5 UNTIL L>LN
                  *
                  *   COMPUTE POWER SPECTRUM BY SQUARING POSITIVE PART OF
                  *   MAGNITUDE
                  *
                   DSPLOPT PS,PSE
006D60 4BF86EC6        LEA PS,A5
006D64 4DF86EDE        LEA PSE,A6
                   TRP14 OUT1CR
006D68 1E3C00E3        MOVE.B #OUT1CR,D7
006D6C 4E4E    TRAP #14
006D6E 207C00050001  MOVE.L #APUOPER,A0
                   STPOIN2.W XRST,A2,XIST,A3
006D74 347C3700        MOVE.W #XRST,A2
006D78 367C4700        MOVE.W #XIST,A3
006D7C 303C0000        MOVE.W #$0,D0
                  FFT6 APUSGWR A2,D0,A0   WRITE REAL PART TO STACK
006D80 2C320000        MOVE.L 0(A2,D0),D6
006D84 3E3C0003        MOVE.W #$3,D7
006D88 1086    @156 MOVE.B D6,(A0)
006D8A E09E    ROR.L #$8,D6
006D8C 51CFFFFA        DBRA D7,@156
                   APU PTOD               PUSH TOS
006D90 13FC0037
     00050011  MOVE.B #PTOD,APUCOM
006D98 4E722300        STOP #$2300
                   APU FMUL               SQUARE REAL PART
006D9C 13FC0012
     00050011  MOVE.B #FMUL,APUCOM
006DA4 4E722300        STOP #$2300
                   APUSGWR A3,D0,A0       WRITE IMAGINARY PART TO STACK
006DA8 2C330000        MOVE.L 0(A3,D0),D6
006DAC 3E3C0003        MOVE.W #$3,D7
006DB0 1086    @159 MOVE.B D6,(A0)
006DB2 E09E    ROR.L #$8,D6
006DB4 51CFFFFA        DBRA D7,@159
                   APU PTOD
006DB8 13FC0037
     00050011  MOVE.B #PTOD,APUCOM
006DC0 4E722300        STOP #$2300
                   APU FMUL               SQUARE IMAGINARY PART
006DC4 13FC0012
     00050011  MOVE.B #FMUL,APUCOM
```

```
006DCC 4E722300       STOP #$2300
              APU FADD                   ADD XR**2 TO XI**2
006DD0 13FC0010
    00050011  MOVE.B #FADD,APUCOM
006DD8 4E722300       STOP #$2300
              APUSGRD A2,D0,A0           STORE POWER SPECTRUM SAMPLE IN XREAL
006DDC 3E3C0003       MOVE.W #$3,D7
006DE0 E19E   @163 ROL.L #$8,D6
006DE2 1C10   MOVE.B (A0),D6
006DE4 51CFFFFA       DBRA D7,@163
006DE8 25860000       MOVE.L D6,0(A2,D0)
006DEC 06400004       ADDI.W #$4,D0
006DF0 0C401000       CMPI.W #4096,D0
006DF4 6D8A   BLT FFT6
006DF6 4E75   RTS                       SKIP CONVOLUTION ROUTINE FOR NOW
             *                          CAN USE ROUTINE BY REMOVING THIS RTS.
             *
             *        CONVOLUTION ROUTINE TO SMOOTH POWER SPECTRA
             *
              DSPLOPT SMPS,SMPSE
006DF8 4BF86EE0       LEA SMPS,A5
006DFC 4DF86F00       LEA SMPSE,A6
              TRP14 OUT1CR
006E00 1E3C00E3       MOVE.B #OUT1CR,D7
006E04 4E4E   TRAP #14
006E06 207C00050001  MOVE.L #APUOPER,A0
006E0C 7008   MOVE.L #8,D0              CONVOLVE WINDOW WIDTH
006E0E 347C3700       MOVE.W #XRST,A2   START OF POWER SPECTRUM
006E12 387C4700       MOVE.W #TMPSTC,A4        TEMPORARY STORAGE POINTER
006E16 584A   ADDQ.W #$4,A2              INCREMENT POWER SPECTRA POINTER
             *
             *
006E18 364A  FLCN1 MOVE.W A2,A3          CONVOLVE INNER LOOP POINTER
006E1A 4281   CLR.L D1                   C(N)=0
006E1C 2400   MOVE.L D0,D2               # OF SAMPLES IN WINDOW
             *
006E1E 2623  FLCN2 MOVE.L -(A3),D3       DECREMENT, THEN LOAD DATA
006E20 B6FC3700       CMP.W #XRST,A3
006E24 6D00005A       BLT FLCN3          ENSURE WINDOW NOT BEFORE XRST.
              APUWR D3,A0                X(N-K)
006E28 3E3C0003       MOVE.W #$3,D7
006E2C 1083   @165 MOVE.B D3,(A0)
006E2E E09B   ROR.L #$8,D3
006E30 51CFFFFA       DBRA D7,@165
              APUWR D2,A0                W(K)
006E34 3E3C0003       MOVE.W #$3,D7
006E38 1082   @166 MOVE.B D2,(A0)
006E3A E09A   ROR.L #$8,D2
006E3C 51CFFFFA       DBRA D7,@166
              APU FLTD
006E40 13FC001C
    00050011  MOVE.B #FLTD,APUCOM
006E48 4E722300       STOP #$2300
              APU FMUL                   X(N-K)W(K)
006E4C 13FC0012
    00050011  MOVE.B #FMUL,APUCOM
006E54 4E722300       STOP #$2300
              APUWR D1,A0
006E58 3E3C0003       MOVE.W #$3,D7
006E5C 1081   @169 MOVE.B D1,(A0)
006E5E E099   ROR.L #$8,D1
```

```
006E60 51CFFFFA       DBRA D7,@169
              APU FADD                     C(N) SUMMATION
006E64 13FC0010
    00050011  MOVE.B #FADD,APUCOM
006E6C 4E722300       STOP #$2300
              APURD A0,D1
006E70 3E3C0003       MOVE.W #$3,D7
006E74 E199   @171 ROL.L #$8,D1
006E76 1210   MOVE.B (A0),D1
006E78 51CFFFFA       DBRA D7,@171
006E7C 51CAFFA0       DBRA D2,FLCN2  CONTINUE FOR ENTIRE WINDOW
              *
006E80 28C1   FLCN3 MOVE.L D1,(A4)+     C(N) FOR SPECIFIC N
006E82 584A   ADDQ.W #$4,A2             INCREMENT POINTER
006E84 B4FC3EFC       CMP.W #XRENH,A2
006E88 6D8E   BLT FLCN1
              *
              *  MOVE DATA BACK TO XRST
              *
006E8A 307C3700       MOVE.W #XRST,A0
006E8E 327C4700       MOVE.W #TMPSTC,A1
006E92 20D9   FLCNE MOVE.L (A1)+,(A0)+     MOVE DATA
006E94 B0FC3EFC       CMP.W #XRENH,A0
006E98 6DF8   BLT FLCNE
006E9A 4E75   RTS
006E9C 20     HAM DC.W '          HAMMING WINDOW'
006EB4 20     HAME DC.W ' '
006EB6 20     FFTM DC.W '          FFT'
006EC4 20     FFTME DC.W ' '
006EC6 20     PS DC.W '          POWER SPECTRUM'
006EDE 20     PSE DC.W ' '
006EE0 20     SMPS DC.W '          SMOOTH POWER SPECTRUM'
006F00 20     SMPSE DC.W ' '
              *
              *
```

```
                   *##############################################################
                   *##############################################################
                   *
                   *              PCG PROCESSING - FREQ DOMAIN
                   *
                   *##############################################################
                   *  THE PCG PROCESSING ROUTINE IS COMPRISED OF THE FOLLOWING
                   *  MODULES:
                   *
                   *  PFMSYS:  MOVE PCG SYSTOLE INTO COMPLEX ARRAY
                   *  FFT:     COMPUTE POWER SPECTRUM OF SYSTOLE
                   *  PFPSYS:  MOVE SYSTOLE POWER SPECTRUM TO TEMP
                   *  PFMDIA:  MOVE DIASTOLE INTO COMPLEX ARRAY
                   *  FFT:     COMPUTE POWER SPECTRUM OF DIASTOLE
                   *  PFEDC:   COMPUTE ENERGY DISTRIBUTION COEFFICIENTS
                   *##############################################################
                   *##############################################################
                   *
006F02 4EB872D4      PCGF JSR BLANK
                    DSPLOPT PFST,PFEN
006F06 4BF86F58            LEA PFST,A5
006F0A 4DF86F7A            LEA PFEN,A6
                    TRP14 OUT1CR
006F0E 1E3C00E3            MOVE.B #OUT1CR,D7
006F12 4E4E         TRAP #14
                    DSPLOPT PFSFT,PFSFTE
006F14 4BF86F7C            LEA PFSFT,A5
006F18 4DF86F8E            LEA PFSFTE,A6
                    TRP14 OUT1CR
006F1C 1E3C00E3            MOVE.B #OUT1CR,D7
006F20 4E4E         TRAP #14
006F22 4EB86FBE            JSR PFMSYS
006F26 4EB86A8A            JSR FFT
                    DSPLOPT PFDFT,PFDFTE
006F2A 4BF86F90            LEA PFDFT,A5
006F2E 4DF86FA2            LEA PFDFTE,A6
                    TRP14 OUT1CR
006F32 1E3C00E3            MOVE.B #OUT1CR,D7
006F36 4E4E         TRAP #14
006F38 4EB8700C            JSR PFPSYS
006F3C 4EB87022            JSR PFMDIA
006F40 4EB86A8A            JSR FFT
                    DSPLOPT PFEST,PFEEN
006F44 4BF86FA4            LEA PFEST,A5
006F48 4DF86FBC            LEA PFEEN,A6
                    TRP14 OUT1CR
006F4C 1E3C00E3            MOVE.B #OUT1CR,D7
006F50 4E4E         TRAP #14
006F52 4EB87062            JSR PFEDC
006F56 4E75         RTS
006F58 50          PFST DC.W 'PCG PROCESSING - FREQUENCY DOMAIN'
006F7A 20          PFEN DC.W ' '
006F7C 20          PFSFT DC.W '     SYSTOLIC FFT'
006F8E 20          PFSFTE DC.W ' '
006F90 20          PFDFT DC.W '     DIASTOLIC FFT'
006FA2 20          PFDFTE DC.W ' '
006FA4 20          PFEST DC.W '     COMPUTE EDC VALUES'
006FBC 20          PFEEN DC.W ' '
```

```
                 ***************************************************************
                 *                         PFMSYS
                 ***************************************************************
                 *
                 *      A0 <--> START OF SYSTOLE
                 *      A1 <--> END OF SYSTOLE
                 *      A2 <--> START OF REAL ARRAY;IMAGINARY ARRAY
                 *      D0 <--> SCRATCH
                 *
                 ***************************************************************
                 *
                 *      CALL FOLLOWING MACROS:
                 *                   GETADD '
                 *
                 ***************************************************************
                 *
                 * THIS ROUTINE MOVES THE SYSTOLIC SEGMENT OF THE PCG INTO THE
                 * REAL ARRAY. THE ARRAY IS APPENDED WITH ZEROS. THE IMAGINARY
                 * ARRAY IS SET TO ZERO.
                 *
                 *
                 PFMSYS GETADD PCGS1A,A0,PCGSTRT
006FBE 30787F08         MOVE.W PCGS1A,A0
006FC2 D0FC0F00         ADD.W #PCGSTRT,A0
                  GETADD PCGS1B,A1,PCGSTRT
006FC6 32787F0A         MOVE.W PCGS1B,A1
006FCA D2FC0F00         ADD.W #PCGSTRT,A1
006FCE 4EB863AC         JSR TWOCOMP          CONVERT PCG TO TWOS COMPLEMENT
                  GETADD PCGS1A,A0,PCGSTRT   START OF SYSTOLE
006FD2 30787F08         MOVE.W PCGS1A,A0
006FD6 D0FC0F00         ADD.W #PCGSTRT,A0
006FDA 3248       MOVE.W A0,A1
006FDC D2F87F0C         ADD.W PCGS2,A1       END OF SYSTOLE
006FE0 347C3700         MOVE.W #XRST,A2      START OF XREAL ARRAY
006FE4 4280       CLR.L D0
006FE6 3018      PFMSYS1 MOVE.W (A0)+,D0     MOVE SYSTOLE IN REAL ARRAY
006FE8 24C0       MOVE.L D0,(A2)+               MOVE DATA
006FEA B0C9       CMP.W A1,A0
006FEC 6DF8       BLT PFMSYS1
006FEE 24FC00000000 PFMSYS2 MOVE.L #0,(A2)+     APPEND WITH ZEROS
006FF4 B4FC4700         CMP.W #XREN,A2
006FF8 6DF4       BLT PFMSYS2
006FFA 347C4700         MOVE.W #XIST,A2
006FFE 24FC00000000 PFMSYS3 MOVE.L #0,(A2)+     SET IMAGINARY ARRAY TO ZERO
007004 B4FC5700         CMP.W #XIEN,A2
007008 6DF4       BLT PFMSYS3
00700A 4E75       RTS
                 *
                 *
```

```
                 ***************************************************************
                 *                                PFPSYS
                 ***************************************************************
                 *
                 *      A2 <--> START OF REAL ARRAY;POWER SPECTRUM
                 *      A3 <--> TEMPORARY STORAGE
                 *      DO <--> COUNTER
                 *
                 ***************************************************************
                 *
                 *      CALL FOLLOWING MACROS:
                 *                   STPOIN2
                 *
                 ***************************************************************
                 *
                 *
                 * THE SYSTLIC POWER SPECTRUM IS RETURNED IN THE FIRST HALF OF
                 * REAL ARRAY. THIS ROUTINE MOVES THE POWER SPECTRUM TO TEMPORA
                 * STORAGE.
                 *
                 *
                 PFPSYS STPOIN2.W XRST,A2,TMPSTB,A3
00700C 347C3700         MOVE.W #XRST,A2
007010 367C5700         MOVE.W #TMPSTB,A3
007014 303C0200         MOVE.W #512,D0
007018 26DA      PFPSYS1 MOVE.L (A2)+,(A3)+     MOVE DATA
00701A 51C8FFFC         DBRA D0,PFPSYS1
00701E 6DF8       BLT PFPSYS1
007020 4E75       RTS
                 *
                 *
```

```
                   *************************************************************
                   *                         PFMDIA
                   *************************************************************
                   *
                   *      A0 <--> START OF DIASTOLE
                   *      A1 <--> END OF DIASTOLE
                   *      A2 <--> START OF REAL ARRAY;IMAGINARY ARRAY
                   *
                   *************************************************************
                   *
                   *      CALL FOLLOWING MACROS:
                   *                   GETADD ''
                   *
                   *************************************************************
                   *
                   * THIS ROUTINE MOVES THE DIASTOLIC SEGMENT OF THE PCG INTO THE
                   * REAL ARRAY. THE ARRAY IS APPENDED WITH ZEROS. THE IMAGINARY
                   * ARRAY IS SET TO ZERO.
                   *
                   *
                   PFMDIA GETADD PCGS1A,A0,PCGSTRT  START OF SYSTOLE
007022 30787F08           MOVE.W PCGS1A,A0
007026 D0FC0F00           ADD.W #PCGSTRT,A0
00702A D0F87F0C           ADD.W PCGS2,A0      START OF DIASTOLE
                    GETADD PCGS1B,A1,PCGSTRT  END OF DIASTOLE
00702E 32787F0A           MOVE.W PCGS1B,A1
007032 D2FC0F00           ADD.W #PCGSTRT,A1
007036 347C3700           MOVE.W #XRST,A2     START OF XREAL ARRAY
00703A 4280         CLR.L D0
00703C 3018        PFMDIA1 MOVE.W (A0)+,D0    MOVE DIASTOLE IN REAL ARRAY
00703E 24C0         MOVE.L D0,(A2)+                MOVE DATA
007040 B0C9         CMP.W A1,A0
007042 6DF8         BLT PFMDIA1
007044 24FC00000000 PFMDIA2 MOVE.L #0,(A2)+    APPEND WITH ZEROS
00704A B4FC4700           CMP.W #XREN,A2
00704E 6DF4         BLT PFMDIA2
007050 347C4700           MOVE.W #XIST,A2
007054 24FC00000000 PFMDIA3 MOVE.L #0,(A2)+    SET IMAGINARY ARRAY TO ZERO
00705A B4FC5700           CMP.W #XIEN,A2
00705E 6DF4         BLT PFMDIA3
007060 4E75         RTS
                   *
                   *
```

```
                   ****************************************************************
                   *                          PFEDC
                   ****************************************************************
                   *
                   *      A2 <--> START OF SIGNAL
                   *      D6 <--> EDC VALUE IS RETURNED IN THIS REGISTER
                   *
                   ****************************************************************
                   *
                   *      CALL FOLLOWING MACROS:              SUBROUTINES:
                   *                                                EDC ''
                   *
                   ****************************************************************
                   *
                   *
                   * COMPUTE EDC VALUES OF SYSTOLIC AND DIASTOLIC SEGMENTS
                   *
                   *
007062 347C5700         PFEDC MOVE.W #TMPSTB,A2
007066 4EB87094          JSR FLEDC
00706A 21C67F1A          MOVE.L D6,EDCFSYS
00706E 347C3700          MOVE.W #XRST,A2
007072 4EB87094          JSR FLEDC
007076 21C67F1E          MOVE.L D6,EDCFDIA
                   *
                    GETADD PCGS1A,A0,PCGSTRT  CONVERT BACK TO 12-BIT NUMBER
00707A 30787F08          MOVE.W PCGS1A,A0
00707E D0FC0F00          ADD.W #PCGSTRT,A0
                    GETADD PCGS1B,A1,PCGSTRT
007082 32787F0A          MOVE.W PCGS1B,A1
007086 D2FC0F00          ADD.W #PCGSTRT,A1
00708A 06580800         PFEDC1 ADD.W #$800,(A0)+     ADD DATA
00708E B0C9     CMP.W A1,A0
007090 6FF8     BLE.S PFEDC1
007092 4E75     RTS
                   *
                   *
```

```
                  ***************************************************************
                  *                        FLEDC
                  ***************************************************************
                  *
                  *     A0 <--> APU OPERAND ENTRY ADDRESS
                  *     A1 <--> APU COMMAND ENTRY ADDRESS
                  *     A2 <--> START OF SIGNAL. WHEN ROUTINE CALLED START OF
                  *             SIGNAL MUST BE IN A2
                  *     D0 <--> N (MULTIPLIER IN NUMERATOR)
                  *     D3 <--> SQUARE ROOT OF VALUE
                  *     D6 <--> FINAL EDC VALUE 32-BIT FIXED POINT
                  *
                  ***************************************************************
                  *
                  *     CALL FOLLOWING MACROS:
                  *      STPOIN2     APUWR ''     APUSGWR ''     APURD      APU
                  *
                  ***************************************************************
                  *
                  *
                  * THIS ROUTINE COMPUTES THE FLOATING POINT EDC VALUE. PARAMETE
                  * ARE ENTERED AS 32-BIT FLOATING POINT. EDC VALUES COMPUTED FO
                  * POSITIVE SIDE OF POWER SPECTRUM. RETURNED VALUE (EDC) IS
                  * CONVERTED TO 32 FIXED POINT AND RESIDES IN DATA REGISTER D6.
                  *
                  *
                  FLEDC STPOIN2.L APUOPER,A0,APUCOM,A1
007094 207C00050001  MOVE.L #APUOPER,A0
00709A 227C00050011  MOVE.L #APUCOM,A1
0070A0 7000     MOVE.L #$0,D0
                   APUWR D0,A0
0070A2 3E3C0003        MOVE.W #$3,D7
0070A6 1080   @185 MOVE.B D0,(A0)
0070A8 E098     ROR.L #$8,D0
0070AA 51CFFFFA        DBRA D7,@185
                   APUWR D0,A0                   MOVE NUMERATOR AND DENOMINATOR ON
0070AE 3E3C0003        MOVE.W #$3,D7
0070B2 1080   @186 MOVE.B D0,(A0)
0070B4 E098     ROR.L #$8,D0
0070B6 51CFFFFA        DBRA D7,@186
                  *                              STACK (INITIALLY ZERO).
0070BA 4241     CLR.W D1                         SIGNAL ARRAY POINTER
0070BC 7401     MOVE.L #$1,D2                    MULTIPLIER (N) IN EDC COMPUTATION
                  FLEDC1 APUSGWR A2,D1,A0        WRITE POWER SPECTRA SAMPLE TO STA
0070BE 2C321000        MOVE.L 0(A2,D1),D6
0070C2 3E3C0003        MOVE.W #$3,D7
0070C6 1086   @187 MOVE.B D6,(A0)
0070C8 E09E     ROR.L #$8,D6
0070CA 51CFFFFA        DBRA D7,@187
                   APU FADD                      UPDATE DENOMINATOR
0070CE 13FC0010
       00050011 MOVE.B #FADD,APUCOM
0070D6 4E722300        STOP #$2300
                   APU XCHF                      EXCHANGE NUM AND DEN ON STACK
0070DA 13FC0019
       00050011 MOVE.B #XCHF,APUCOM
0070E2 4E722300        STOP #$2300
                   APUWR D2,A0                   WRITE N TO STACK
0070E6 3E3C0003        MOVE.W #$3,D7
0070EA 1082   @190 MOVE.B D2,(A0)
0070EC E09A     ROR.L #$8,D2
```

```
2210 0070EE 51CFFFFA       DBRA D7,@190
2211               APU FLTD                        FLOATING POINT
2211 0070F2 13FC001C
        00050011  MOVE.B #FLTD,APUCOM
2211 0070FA 4E722300       STOP #$2300
2212               APUSGWR A2,D1,A0                WRITE POWER SPECTRA VALUE TO STAC
2212 0070FE 2C321000       MOVE.L 0(A2,D1),D6
2212 007102 3E3C0003       MOVE.W #$3,D7
2212 007106 1086    @192 MOVE.B D6,(A0)
2212 007108 E09E    ROR.L #$8,D6
2212 00710A 51CFFFFA       DBRA D7,@192
2213               APU FMUL                        N * S(N)
2213 00710E 13FC0012
        00050011  MOVE.B #FMUL,APUCOM
2213 007116 4E722300       STOP #$2300
2214               APU FADD                        UPDATE NUMERATOR
2214 00711A 13FC0010
        00050011  MOVE.B #FADD,APUCOM
2214 007122 4E722300       STOP #$2300
2215               APU XCHF                        EXCHANGE NUMAND DEN
2215 007126 13FC0019
        00050011  MOVE.B #XCHF,APUCOM
2215 00712E 4E722300       STOP #$2300
2216 007132 5282    ADD.L #$1,D2                   INCREMENT MULTIPLIER
2217 007134 5841    ADD.W #$4,D1                   INCREMENT SIGNAL ARRAY POINTER
2218 007136 0C410800       CMP.W #2048,D1
2219 00713A 6D82    BLT FLEDC1                     CONTINUE UNTIL ENTIRE SPECTRA COM
2220               APU FDIV                        NUMERATOR/DENOMINATOR = EDC
2220 00713C 13FC0013
        00050011  MOVE.B #FDIV,APUCOM
2220 007144 4E722300       STOP #$2300
2221               APU FIXD                        CONVERT TO 32-BIT FIXED
2221 007148 13FC001E
        00050011  MOVE.B #FIXD,APUCOM
2221 007150 4E722300       STOP #$2300
2222               APURD A0,D6                     D6 CONATINS EDC VALUE
2222 007154 3E3C0003       MOVE.W #$3,D7
2222 007158 E19E    @198 ROL.L #$8,D6
2222 00715A 1C10    MOVE.B (A0),D6
2222 00715C 51CFFFFA       DBRA D7,@198
2223 007160 4E75    RTS
2224               *
2225               *
```

```
                    *#############################################################
                    *
                    *             D/A CONVERSION ROUTINES
                    *
                    *        1) NORMALIZE ALL TRANSFORM DATA TO BE CONVERTED
                    *           WITH 12 BIT D/A.
                    *        2) DISPLAY OPTIONS MENU
                    *        3) SELECT OPTION
                    *        4) PERFORM CONVERSION
                    *        5) GO TO 2)
                    *
                    *#############################################################
                    *#############################################################
                    *
                    *
                    *  FIND MAXIMUM VALUE IN ARRAY... MAX RETURNED IN D0
                    *  A2 <--> START      A3 <--> END
                    *
007162 4280         DAMAX CLR.L D0               CLEAR INITIAL MAXIMUM
007164 221A         DAMAX1 MOVE.L (A2)+,D1
007166 B081          CMP.L D1,D0                 CHECK CURRENT VALUE AGAINST MAX
007168 6C02          BGE.S DAMAX2
00716A 2001          MOVE.L D1,D0                UPDATE CURRENT MAX
00716C B4CB         DAMAX2 CMP.W A3,A2
00716E 6FF4          BLE.S DAMAX1                SEARCH ENTIRE ARRAY
007170 4E75          RTS
                    *
                    *
                    *  NORMALIZE ALL DATA IN ARRAY TO MAXIMUM VALUE OF 4095 (FFF)
                    *  A0,A1 <--> APU ADDRESSES    A2,A3 <--> START AND END
                    *  D0 <--> MAXIMUM VALUE IN ARRAY
                    *
                    *
                    DANORM APUWR D0,A0
007172 3E3C0003            MOVE.W #$3,D7
007176 1080         @199 MOVE.B D0,(A0)
007178 E098          ROR.L #$8,D0
00717A 51CFFFFA            DBRA D7,@199
                     APU FLTD
00717E 13FC001C
       00050011      MOVE.B #FLTD,APUCOM
007186 4E722300            STOP #$2300
                     APURD A0,D0                  MAX VALUE - FLOATING POINT
00718A 3E3C0003            MOVE.W #$3,D7
00718E E198         @201 ROL.L #$8,D0
007190 1010          MOVE.B (A0),D0
007192 51CFFFFA            DBRA D7,@201
007196 223C0CFFF000  MOVE.L #$0CFFF000,D1         4095 - FLOATING POINT
00719C 2412         DANORM1 MOVE.L (A2),D2        SAMPLE VALUE
                     APUWR D2,A0
00719E 3E3C0003            MOVE.W #$3,D7
0071A2 1082         @202 MOVE.B D2,(A0)
0071A4 E09A          ROR.L #$8,D2
0071A6 51CFFFFA            DBRA D7,@202
                     APU FLTD
0071AA 13FC001C
       00050011      MOVE.B #FLTD,APUCOM
0071B2 4E722300            STOP #$2300
                     APUWR D0,A0
0071B6 3E3C0003            MOVE.W #$3,D7
0071BA 1080         @204 MOVE.B D0,(A0)
```

```
0071BC E098    ROR.L #$8,D0
0071BE 51CFFFFA        DBRA D7,@204
               APU FDIV                          VALUE/MAX
0071C2 13FC0013
       00050011  MOVE.B #FDIV,APUCOM
0071CA 4E722300        STOP #$2300
               APUWR D1,A0
0071CE 3E3C0003        MOVE.W #$3,D7
0071D2 1081   @206 MOVE.B D1,(A0)
0071D4 E099    ROR.L #$8,D1
0071D6 51CFFFFA        DBRA D7,@206
               APU FMUL                          (VALUE/MAX)*4095
0071DA 13FC0012
       00050011  MOVE.B #FMUL,APUCOM
0071E2 4E722300        STOP #$2300
               APU FIXD                           FIXED POINT
0071E6 13FC001E
       00050011  MOVE.B #FIXD,APUCOM
0071EE 4E722300        STOP #$2300
               APURD A0,D2
0071F2 3E3C0003        MOVE.W #$3,D7
0071F6 E19A   @209 ROL.L #$8,D2
0071F8 1410    MOVE.B (A0),D2
0071FA 51CFFFFA        DBRA D7,@209
0071FE 24C2    MOVE.L D2,(A2)+                       MOVE DATA
007200 B4CB    CMP.W A3,A2
007202 6F98    BLE DANORM1                       NORMALIZE ENTIRE ARRAY
007204 4E75    RTS
              *
              *
              *  NORMALIZE ECG, CAROTID, AND PCG TRANSFORMS... DIVIDE
              *  ALL SAMPLES BY THE MAX AND MULTIPLY BY 4095.
              *  A0,A1 <--> APU ADDRESSES
              *  A2,A3 <--> START AND END OF ARRAY
              *
              *
              DAFIX STPOIN2.L APUOPER,A0,APUCOM,A1
007206 207C00050001  MOVE.L #APUOPER,A0
00720C 227C00050011  MOVE.L #APUCOM,A1
               STPOIN2.W TRECGST,A2,TRECGCHK,A3
007212 347C1B00        MOVE.W #TRECGST,A2
007216 367C20F0        MOVE.W #TRECGCHK,A3
00721A 4EB87162        JSR DAMAX
00721E 347C1B00        MOVE.W #TRECGST,A2
007222 4EB87172        JSR DANORM
               STPOIN2.W TRCARST,A2,TRCAREN,A3
007226 347C2100        MOVE.W #TRCARST,A2
00722A 367C2300        MOVE.W #TRCAREN,A3
00722E 4EB87162        JSR DAMAX
007232 347C2100        MOVE.W #TRCARST,A2
007236 4EB87172        JSR DANORM
00723A 347C2300        MOVE.W #PCGENST,A2
00723E 36787F10        MOVE.W PCGENDE,A3
007242 4EB87162        JSR DAMAX
007246 347C2300        MOVE.W #PCGENST,A2
00724A 4EB87172        JSR DANORM
00724E 6000003C        BRA DAFIXB
              *
              *
              * TAKE SQUARE ROOT OF POWER SPECTRUM -> FOURIER TRANSFORM
              * IN 32-BIT FIXED POINT (FOR DISPLAY PURPOSES)
```

```
                     *
007252 2412   PSTOFT MOVE.L (A2),D2
                APUWR D2,A0
007254 3E3C0003        MOVE.W #$3,D7
007258 1082     @213 MOVE.B D2,(A0)
00725A E09A     ROR.L #$8,D2
00725C 51CFFFFA        DBRA D7,@213
                APU SQRT
007260 13FC0001
    00050011    MOVE.B #SQRT,APUCOM
007268 4E722300        STOP #$2300
                APU FIXD
00726C 13FC001E
    00050011    MOVE.B #FIXD,APUCOM
007274 4E722300        STOP #$2300
                APURD A0,D2
007278 3E3C0003        MOVE.W #$3,D7
00727C E19A     @216 ROL.L #$8,D2
00727E 1410     MOVE.B (A0),D2
007280 51CFFFFA        DBRA D7,@216
007284 24C2     MOVE.L D2,(A2)+                 MOVE DATA
007286 B4CB     CMP.W A3,A2
007288 6FC8     BLE PSTOFT
00728A 4E75     RTS
               *
               *  NORMALIZE FREQUENCY DOMAIN DATA. TAKE SQUARE ROOT OF POWER
               *  SPECTRUM TO GET FT - CONVERT TO 32-BIT FIXED POINT. THE
               *  NORMALIZE DATA.... DIVIDE SAMPLES BY MAX AND MULTIPLY BY
               *  4095.   A0,A2 <--> APU ADDRESSES,  A2,A3 <--> START AND END
               *
               DAFIXB STPOIN2.L APUOPER,A0,APUCOM,A1
00728C 207C00050001  MOVE.L #APUOPER,A0
007292 227C00050011  MOVE.L #APUCOM,A1
                STPOIN2.W XRST,A2,XRENH,A3
007298 347C3700        MOVE.W #XRST,A2
00729C 367C3EFC        MOVE.W #XRENH,A3
0072A0 4EB87252        JSR PSTOFT
0072A4 347C3700        MOVE.W #XRST,A2
0072A8 4EB87162        JSR DAMAX
0072AC 347C3700        MOVE.W #XRST,A2
0072B0 4EB87172        JSR DANORM
                STPOIN2.W TMPSTB,A2,TMPENB,A3
0072B4 347C5700        MOVE.W #TMPSTB,A2
0072B8 367C5EFC        MOVE.W #TMPENB,A3
0072BC 4EB87252        JSR PSTOFT
0072C0 347C5700        MOVE.W #TMPSTB,A2
0072C4 4EB87162        JSR DAMAX
0072C8 347C5700        MOVE.W #TMPSTB,A2
0072CC 4EB87172        JSR DANORM
0072D0 60000016        BRA DA
               *
               *
               *  CLEAR SCREEN
               *
               BLANK SPACE
0072D4 4BF872E4        LEA SPAC,A5
0072D8 4DF872E6        LEA SPCE,A6
                TRP14 OUT1CR
0072DC 1E3C00E3        MOVE.B #OUT1CR,D7
0072E0 4E4E     TRAP #14
0072E2 4E75     RTS
```

```
2340 0072E4 20      SPAC DC.W ' '
2341 0072E6 20      SPCE DC.W ' '
2342                *
2343 0072E8 33FC0800
        00040000 DA MOVE.W #$800,DACH1
2344 0072F0 33FC0000
        00040004  MOVE.W #0,DACH2
2345 0072F8 13FC0000
        00010021  MOVE.B #HLTIMR,TCR        HALT TIMER USING TCR HALT CODE
2346 007300 343C001E       MOVE.W #30,D2
2347                DAMEN SPACE
2347 007304 4BF872E4       LEA SPAC,A5
2347 007308 4DF872E6       LEA SPCE,A6
2347                 TRP14 OUT1CR
2347 00730C 1E3C00E3       MOVE.B #OUT1CR,D7
2347 007310 4E4E     TRAP #14
2348 007312 51CAFFF0       DBRA D2,DAMEN
2349                *
2350                * DISPLAY OPTIONS
2351                *
2352                DAOPT DSPLOPT MSG,MSGE
2352 007316 4BF875F6       LEA MSG,A5
2352 00731A 4DF87628       LEA MSGE,A6
2352                 TRP14 OUT1CR
2352 00731E 1E3C00E3       MOVE.B #OUT1CR,D7
2352 007322 4E4E     TRAP #14
2353 007324 4EB872D4       JSR BLANK
2354 007328 4EB872D4       JSR BLANK
2355                 DSPLOPT OPT1,OPT1E
2355 00732C 4BF8762A       LEA OPT1,A5
2355 007330 4DF8765E       LEA OPT1E,A6
2355                 TRP14 OUT1CR
2355 007334 1E3C00E3       MOVE.B #OUT1CR,D7
2355 007338 4E4E     TRAP #14
2356                 DSPLOPT OPT2,OPT2E
2356 00733A 4BF87660       LEA OPT2,A5
2356 00733E 4DF8769A       LEA OPT2E,A6
2356                 TRP14 OUT1CR
2356 007342 1E3C00E3       MOVE.B #OUT1CR,D7
2356 007346 4E4E     TRAP #14
2357                 DSPLOPT OPT3,OPT3E
2357 007348 4BF8769C       LEA OPT3,A5
2357 00734C 4DF876D6       LEA OPT3E,A6
2357                 TRP14 OUT1CR
2357 007350 1E3C00E3       MOVE.B #OUT1CR,D7
2357 007354 4E4E     TRAP #14
2358                 DSPLOPT OPT4,OPT4E
2358 007356 4BF876D8       LEA OPT4,A5
2358 00735A 4DF87718       LEA OPT4E,A6
2358                 TRP14 OUT1CR
2358 00735E 1E3C00E3       MOVE.B #OUT1CR,D7
2358 007362 4E4E     TRAP #14
2359                 DSPLOPT OPT5,OPT5E
2359 007364 4BF8771A       LEA OPT5,A5
2359 007368 4DF8775C       LEA OPT5E,A6
2359                 TRP14 OUT1CR
2359 00736C 1E3C00E3       MOVE.B #OUT1CR,D7
2359 007370 4E4E     TRAP #14
2360                 DSPLOPT OPT6,OPT6E
2360 007372 4BF8775E       LEA OPT6,A5
2360 007376 4DF8779C       LEA OPT6E,A6
```

```
2360                    TRP14 OUT1CR
2360 00737A 1E3C00E3          MOVE.B #OUT1CR,D7
2360 00737E 4E4E        TRAP #14
2361                    DSPLOPT OPT7,OPT7E
2361 007380 4BF8779E          LEA OPT7,A5
2361 007384 4DF877E0          LEA OPT7E,A6
2361                    TRP14 OUT1CR
2361 007388 1E3C00E3          MOVE.B #OUT1CR,D7
2361 00738C 4E4E        TRAP #14
2362                    DSPLOPT OPT8,OPT8E
2362 00738E 4BF877E2          LEA OPT8,A5
2362 007392 4DF87826          LEA OPT8E,A6
2362                    TRP14 OUT1CR
2362 007396 1E3C00E3          MOVE.B #OUT1CR,D7
2362 00739A 4E4E        TRAP #14
2363                    DSPLOPT OPT9,OPT9E
2363 00739C 4BF87828          LEA OPT9,A5
2363 0073A0 4DF87862          LEA OPT9E,A6
2363                    TRP14 OUT1CR
2363 0073A4 1E3C00E3          MOVE.B #OUT1CR,D7
2363 0073A8 4E4E        TRAP #14
2364                    DSPLOPT OPTA,OPTAE
2364 0073AA 4BF87864          LEA OPTA,A5
2364 0073AE 4DF87898          LEA OPTAE,A6
2364                    TRP14 OUT1CR
2364 0073B2 1E3C00E3          MOVE.B #OUT1CR,D7
2364 0073B6 4E4E        TRAP #14
2365                    DSPLOPT OPTB,OPTBE
2365 0073B8 4BF8789A          LEA OPTB,A5
2365 0073BC 4DF878D4          LEA OPTBE,A6
2365                    TRP14 OUT1CR
2365 0073C0 1E3C00E3          MOVE.B #OUT1CR,D7
2365 0073C4 4E4E        TRAP #14
2366                    DSPLOPT OPTC,OPTCE
2366 0073C6 4BF878D6          LEA OPTC,A5
2366 0073CA 4DF8790E          LEA OPTCE,A6
2366                    TRP14 OUT1CR
2366 0073CE 1E3C00E3          MOVE.B #OUT1CR,D7
2366 0073D2 4E4E        TRAP #14
2367                    DSPLOPT OPTF,OPTFE
2367 0073D4 4BF87910          LEA OPTF,A5
2367 0073D8 4DF87944          LEA OPTFE,A6
2367                    TRP14 OUT1CR
2367 0073DC 1E3C00E3          MOVE.B #OUT1CR,D7
2367 0073E0 4E4E        TRAP #14
2368                   *
2369                   *  ENTER OPTION
2370                   *
2371 0073E2 4EB872D4          JSR BLANK
2372 0073E6 4EB872D4          JSR BLANK
2373 0073EA 4BF87946        NTROPT LEA ENTR,A5
2374 0073EE 4DF8795C          LEA ENTRE,A6
2375                    TRP14 OUTPUT
2375 0073F2 1E3C00F3          MOVE.B #OUTPUT,D7
2375 0073F6 4E4E        TRAP #14
2376                    TRP14 INCHE                   WAIT FOR SINGLE CHARACTER INPUT
2376 0073F8 1E3C00F7          MOVE.B #INCHE,D7
2376 0073FC 4E4E        TRAP #14
2377                   *
2378 0073FE 0C000031          CMP.B #$31,D0
2379 007402 67000068          BEQ ECGCAR
```

```
007406 0C000032      CMP.B #$32,D0
00740A 670000FC      BEQ PCGETR
00740E 0C000033      CMP.B #$33,D0
007412 6700007E      BEQ ECGTR
007416 0C000034      CMP.B #$34,D0
00741A 6700009C      BEQ CARETR
00741E 0C000035      CMP.B #$35,D0
007422 670000BA      BEQ CARTR
007426 0C000036      CMP.B #$36,D0
00742A 670000FE      BEQ PCGCTR
00742E 0C000037      CMP.B #$37,D0
007432 6700011C      BEQ PCGSYS
007436 0C000038      CMP.B #$38,D0
00743A 6700013C      BEQ PCGDIA
00743E 0C000039      CMP.B #$39,D0
007442 67000162      BEQ SYSFT
007446 0C000041      CMP.B #$41,D0
00744A 67000182      BEQ SYSPS
00744E 0C000042      CMP.B #$42,D0
007452 67000166      BEQ DIAFT
007456 0C000043      CMP.B #$43,D0
00745A 67000186      BEQ DIAPS
00745E 0C000045      CMP.B #$45,D0
007462 66000004      BNE NTROPT1
007466 4E75   RTS
007468 6000FE7E     NTROPT1 BRA DA
              *
              *
              *
              *
              *****************************************************************
              *                   D/A DISPLAY ROUTINES
              *****************************************************************
              *      A0 <--> STARTING ADDRESS CHANNEL 1
              *      A1 <--> STARTING ADDRESS CHANNEL 2
              *      A2 <--> ENDING ADDRESS CHANNEL 1
              *      A3 <--> ENDING ADDRESS CHANNEL 2
              *      D0 <--> SAMPLING PERIOD
              *      D1 <--> WORD/LONG WORD INDICATOR
              *      D3 <--> SAMPLING CODE (FOR CHANNEL 2)
              *****************************************************************
              * DISPLAY ECG AND CAROTID PULSE ON CHANNELS 1 AND 2
              *
00746C 307C0900     ECGCAR MOVE.W #ECGSTRT,A0
007470 327C0C00      MOVE.W #CARSTRT,A1
007474 347C0C00      MOVE.W #ECGEND,A2
007478 367C0F00      MOVE.W #CAREND,A3
00747C 203C000001E8  MOVE.L #ADWNSMP,D0
007482 163C0000      MOVE.B #0,D3
007486 123C0000      MOVE.B #$0,D1
00748A 4EB8795E      JSR DA2CH
00748E 6000FE58      BRA DA
              *
              * DISPLAY ECG AND ECG TRANSFORM ON CHANNELS 1 AND 2
              *
007492 307C0900     ECGTR MOVE.W #ECGSTRT,A0
007496 327C1B00      MOVE.W #TRECGST,A1
00749A 347C0C00      MOVE.W #ECGEND,A2
00749E 367C20F0      MOVE.W #TRECGCHK,A3
0074A2 203C000001E8  MOVE.L #ADWNSMP,D0
0074A8 163C0000      MOVE.B #0,D3
```

```
0074AC 123C0001       MOVE.B #1,D1
0074B0 4EB8795E       JSR DA2CH
0074B4 6000FE32       BRA DA
                     *
                     * DISPLAY CAROTID AND ECG TRANSFORM ON CHANNELS 1 AND 2
                     *
0074B8 307C0C00      CARETR MOVE.W #CARSTRT,A0
0074BC 327C1B00       MOVE.W #TRECGST,A1
0074C0 347C0F00       MOVE.W #CAREND,A2
0074C4 367C20F0       MOVE.W #TRECGCHK,A3
0074C8 203C000001E8   MOVE.L #ADWNSMP,D0
0074CE 163C0000       MOVE.B #0,D3
0074D2 123C0001       MOVE.B #1,D1
0074D6 4EB8795E       JSR DA2CH
0074DA 6000FE0C       BRA DA
                     *
                     * DISPLAY CAROTID AND CAROTID TRANSFORM ON CHANNELS 1 AND 2
                     *
0074DE 307C0C00      CARTR MOVE.W #CARSTRT,A0
0074E2 D0F87F00       ADD.W QRS1,A0
0074E6 327C2100       MOVE.W #TRCARST,A1
0074EA 347C0F00       MOVE.W #CAREND,A2
0074EE 367C2300       MOVE.W #TRCAREN,A3
0074F2 203C000001E8   MOVE.L #ADWNSMP,D0
0074F8 163C0000       MOVE.B #0,D3
0074FC 123C0001       MOVE.B #1,D1
007500 4EB8795E       JSR DA2CH
007504 6000FDE2       BRA DA
                     *
                     * DISPLAY PCG AND ECG TRANSFORM ON CHANNELS 1 AND 2
                     *
007508 307C0F00      PCGETR MOVE.W #PCGSTRT,A0
00750C 327C1B00       MOVE.W #TRECGST,A1
007510 347C1B00       MOVE.W #PCGEND,A2
007514 367C20F0       MOVE.W #TRECGCHK,A3
007518 707A    MOVE.L #ADSMPL,D0
00751A 123C0001       MOVE.B #1,D1
00751E 163C0003       MOVE.B #3,D3
007522 4EB8795E       JSR DA2CH
007526 6000FDC0       BRA DA
                     *
                     * DISPLAY PCG AND CAROTID TRANSFORM AN CHANNELS 1 AND 2
                     *
00752A 307C0F00      PCGCTR MOVE.W #PCGSTRT,A0
00752E D0F87F08       ADD.W PCGS1A,A0
007532 327C2100       MOVE.W #TRCARST,A1
007536 347C1B00       MOVE.W #PCGEND,A2
00753A 367C2300       MOVE.W #TRCAREN,A3
00753E 707A    MOVE.L #ADSMPL,D0
007540 123C0001       MOVE.B #1,D1
007544 163C0003       MOVE.B #3,D3
007548 4EB8795E       JSR DA2CH
00754C 6000FD9A       BRA DA
                     *
                     * DISPLAY SYSTOLIC PCG AND ENERGY ON CHANNELS 1 AND 2
                     *
007550 307C0F00      PCGSYS MOVE.W #PCGSTRT,A0
007554 D0F87F08       ADD.W PCGS1A,A0
007558 327C2300       MOVE.W #PCGENST,A1
00755C 3448    MOVE.W A0,A2
00755E D4F87F0C       ADD.W PCGS2,A2
```

```
007562 36787F0E       MOVE.W PCGENSE,A3
007566 707A    MOVE.L #ADSMPL,D0
007568 123C0001       MOVE.B #1,D1
00756C 163C0000       MOVE.B #0,D3
007570 4EB8795E       JSR DA2CH
007574 6000FD72       BRA DA
                *
                * DISPLAY DIASTOLIC PCG AND ENERGY CURVE ON CHANNELS 1 AND 2
                *
007578 307C0F00      PCGDIA MOVE.W #PCGSTRT,A0
00757C D0F87F08       ADD.W PCGS1A,A0
007580 D0F87F0C       ADD.W PCGS2,A0
007584 32787F0E       MOVE.W PCGENSE,A1
007588 347C0F00       MOVE.W #PCGSTRT,A2
00758C 36787F10       MOVE.W PCGENDE,A3
007590 D4F87F0A       ADD.W PCGS1B,A2
007594 707A    MOVE.L #ADSMPL,D0
007596 123C0001       MOVE.B #1,D1
00759A 163C0000       MOVE.B #0,D3
00759E 4EB8795E       JSR DA2CH
0075A2 6000FD44       BRA DA
                *
                * DISPLAY SYSTOLE FOURIER TRANSFORM ON CHANNEL 1
                *
                SYSFT STPOIN2.W TMPSTB,A1,TMPENB,A3
0075A6 327C5700       MOVE.W #TMPSTB,A1
0075AA 367C5EFC       MOVE.W #TMPENB,A3
0075AE 123C0000       MOVE.B #$0,D1
0075B2 4EB879D0       JSR DA2CHFT
0075B6 6000FD30       BRA DA
                *
                * DISPLAY DIASTOLE FOURIER TRANSFORM ON CHANNEL 1
                *
                DIAFT STPOIN2.W XRST,A1,XRENH,A3
0075BA 327C3700       MOVE.W #XRST,A1
0075BE 367C3EFC       MOVE.W #XRENH,A3
0075C2 123C0000       MOVE.B #$0,D1
0075C6 4EB879D0       JSR DA2CHFT
0075CA 6000FD1C       BRA DA
                *
                * DISPLAY SYSTOLE POWER SPECTRUM ON CHANNEL 1
                *
                SYSPS STPOIN2.W TMPSTB,A1,TMPENB,A3
0075CE 327C5700       MOVE.W #TMPSTB,A1
0075D2 367C5EFC       MOVE.W #TMPENB,A3
0075D6 123C0001       MOVE.B #$1,D1
0075DA 4EB879D0       JSR DA2CHFT
0075DE 6000FD08       BRA DA
                *
                * DISPLAY DIASTOLE POWER SPECTRUM ON CHANNEL 1
                *
                DIAPS STPOIN2.W XRST,A1,XRENH,A3
0075E2 327C3700       MOVE.W #XRST,A1
0075E6 367C3EFC       MOVE.W #XRENH,A3
0075EA 123C0001       MOVE.B #$1,D1
0075EE 4EB879D0       JSR DA2CHFT
0075F2 6000FCF4       BRA DA
                *
0075F6 20      MSG DC.W '  EDCTS,EDCTD,EDCFS,EDCFD IN $7F16,7F1B,7FIA,7F1E'
007628 20      MSGE DC.W ' '
00762A 20      OPT1 DC.W '    1. ECG - CHANNEL 1, CAROTID - CHANNEL 2 @ 256 HZ
```

```
00765E 20    OPT1E DC.W ' '
007660 20    OPT2 DC.W '   2. PCG - CHANNEL 1, ECG TRANSFORM - CHANNEL 2'
007690 20     DC.W ' @ 1024 HZ'
00769A 20    OPT2E DC.W ' '
00769C 20    OPT3 DC.W '   3. ECG - CHANNEL 1, ECG TRANSFORM - CHANNEL 2 @
0076D6 20    OPT3E DC.W ' '
0076D8 20    OPT4 DC.W '   4. CAROTID - CHANNEL 1, ECG TRANSFORM - CHANNEL2
00770C 20     DC.W '  @  256 HZ'
007718 20    OPT4E DC.W ' '
00771A 20    OPT5 DC.W '   5. CAROTID - CHANNEL 1, CAROTID TRANSFORM - CHAN
007752 20     DC.W ' @ 256 HZ'
00775C 20    OPT5E DC.W ' '
00775E 20    OPT6 DC.W '   6. PCG - CHANNEL 1, CAROTID TRANSFORM - CHANNEL
007792 20     DC.W ' @ 1024 HZ'
00779C 20    OPT6E DC.W ' '
00779E 20    OPT7 DC.W '   7. PCG(SYSTOLE) - CHANNEL 1, ENERGY CURVE - CHAN
0077D6 20     DC.W ' @ 1024 HZ'
0077E0 20    OPT7E DC.W ' '
0077E2 20    OPT8 DC.W '   8. PCG(DIASTOLE) - CHANNEL 1, ENERGY CURVE - CHA
00781C 20     DC.W ' @ 1024 HZ'
007826 20    OPT8E DC.W ' '
007828 20    OPT9 DC.W '   9. PCG(SYSTOLE) FOURIER TRANSFORM - CHANNEL 1 @
007862 20    OPT9E DC.W ' '
007864 20    OPTA DC.W '   A. PCG(SYSTOLE) POWER SPECTRUM - CHANNEL 1 500 H
007898 20    OPTAE DC.W ' '
00789A 20    OPTB DC.W '   B. PCG(DIASTOLE) FOURIER TRANSFORM - CHANNEL 1 @
0078D4 20    OPTBE DC.W ' '
0078D6 20    OPTC DC.W '   C. PCG(DIASTOLE) POWER SPECTRUM - CHANNEL 1 @ 50
00790E 20    OPTCE DC.W ' '
007910 20    OPTF DC.W '   E. CONTINUE WITH FREQUENCY ANALYSIS/END PROGRAM
007944 20    OPTFE DC.W ' '
007946 20    ENTR DC.W ' ENTER CHOICE (1-E): '
00795C 20    ENTRE DC.W ' '
```

```
                    *****************************************************************
                    *                             DA2CH
                    *****************************************************************
                    *     A0,A2 <--> START,END ADDRESS - CHANNEL 1 DATA
                    *     A1,A3 <--> START,END ADDRESS - CHANNEL 2 DATA
                    *     D0 <--> SAMPLING PERIOD
                    *     D1 <--> WORD/LONG WORD INDICATOR
                    *     D2 <--> SAMPLE VALUE
                    *     D3 <--> SAMPLING CODE
                    *****************************************************************
                    *     CALLS FOLLOWING MACROS:
                    *           DATIMER
                    *           TSTIMR
                    *****************************************************************
                    * THIS SUBROUTINE SENDS DATA (12 BITS) TO D/A CHANNELS 1 & 2.
                    * FOR CASES WHEN TWO SIGNALS WITH DIFFERENT
                    * SAMPLING RATES ARE DISPLAYED, THE SIGNAL WITH THE LARGER
                    * SAMPLING PERIOD MUST GO TO CHANNEL 2 USING THE SAMPLING CODE
                    * THE SAMPLING PERIOD OF THE DATA GOING TO CHANNEL 2 MUST BE
                    * A FACTOR OF THE SAMPLING PERIOD OF THE DATA GOING TO CHANNEL
                    * 1. SAMPLING CODE = ((S.P.1./S.P.2)-1).
                    * D1 IS THE WORD (0)/LONGWORD (1) INDICATOR.
                    *                            INITIALIZE AND START TIMER
                    DA2CH DATIMER                SAMPLING PERIOD MUST BE IN D0
                     STPOIN2.L CNTR,A5,CPR,A6
00795E 2A7C0001002D  MOVE.L #CNTR,A5
007964 2C7C00010025  MOVE.L #CPR,A6
00796A 2E00      MOVE.L D0,D7
00796C 0FCE0000      MOVEP.L D7,0(A6)
007970 13FC0001
     00010035    MOVE.B #1,TSR
007978 13FC0001
     00010021    MOVE.B #STRTIMR,TCR
007980 4204      CLR.B D4
007982 33D800040000 DASIG MOVE.W (A0)+,DACH1
007988 B2CB      CMP.W A3,A1
00798A 6D00000A      BLT DACA
00798E 343C0000      MOVE.W #$0,D2
007992 6000001E      BRA DAC2
007996 0C040000      DACA CMP.B #0,D4
00799A 67000008      BEQ DAC1
00799E 5304      SUBQ.B #1,D4
0079A0 60000010      BRA DAC2
0079A4 1803    DAC1 MOVE.B D3,D4
0079A6 0C010001      CMPI.B #$1,D1
0079AA 66000004      BNE DAC3
0079AE 5449      ADD.W #$2,A1
0079B0 3419    DAC3 MOVE.W (A1)+,D2
0079B2 33C200040004 DAC2 MOVE.W D2,DACH2
                 TSTIMR A2,A0
0079B8 0C390000
     00010035 @243 CMP.B #0,TSR
0079C0 67F6      BEQ @243
0079C2 13FC0001
     00010035    MOVE.B #1,TSR
0079CA B0CA      CMP.W A2,A0
0079CC 6DB4      BLT DASIG
0079CE 4E75      RTS
                 *
```

```
                 ***********************************************************
                 *                                 DA2CHFT
                 ***********************************************************
                 *      A1 <--> START OF FT - CHANNEL 2 DATA; PS
                 *      A3 <--> END  OF  FT - CHANNEL 2 DATA; PS
                 *      A4 <--> APU OPERAND ENTRY ADDRESS
                 *      D0 <--> SAMPLING PERIOD (800 HZ)
                 *      D1 <--> FT/PS INDICATOR
                 *      D2 <--> SAMPLE VALUE          D3 <--> 4095 FLOATING
                 ***********************************************************
                 *      CALLS FOLLOWING MACROS:
                 *            DATIMER           TSTIMR
                 ***********************************************************
                 *
                 * DISPLAY FOURIER TRANSFORM ON CHANNEL 2
                 *
0079D0 287C00050001 DA2CHFT MOVE.L #APUOPER,A4
0079D6 263C0CFFF000  MOVE.L #$0CFFF000,D3
0079DC 203C0000009C  MOVE.L #156,D0
                  DATIMER
                  STPOIN2.L CNTR,A5,CPR,A6
0079E2 2A7C0001002D  MOVE.L #CNTR,A5
0079E8 2C7C00010025  MOVE.L #CPR,A6
0079EE 2E00   MOVE.L D0,D7
0079F0 0FCE0000       MOVEP.L D7,0(A6)
0079F4 13FC0001
       00010035  MOVE.B #1,TSR
0079FC 13FC0001
       00010021  MOVE.B #STRTIMR,TCR
007A04 2419  DA2CHFT1 MOVE.L (A1)+,D2
007A06 0C010001       CMPI.B #$1,D1
007A0A 66000062       BNE DA2CHFT2
                  APUWR D2,A4
007A0E 3E3C0003       MOVE.W #$3,D7
007A12 1882  @245 MOVE.B D2,(A4)
007A14 E09A   ROR.L #$8,D2
007A16 51CFFFFA       DBRA D7,@245
                  APU FLTD
007A1A 13FC001C
       00050011  MOVE.B #FLTD,APUCOM
007A22 4E722300       STOP #$2300
                  APU PTOD
007A26 13FC0037
       00050011  MOVE.B #PTOD,APUCOM
007A2E 4E722300       STOP #$2300
                  APU FMUL
007A32 13FC0012
       00050011  MOVE.B #FMUL,APUCOM
007A3A 4E722300       STOP #$2300
                  APUWR D3,A4
007A3E 3E3C0003       MOVE.W #$3,D7
007A42 1883  @249 MOVE.B D3,(A4)
007A44 E09B   ROR.L #$8,D3
007A46 51CFFFFA       DBRA D7,@249
                  APU FDIV
007A4A 13FC0013
       00050011  MOVE.B #FDIV,APUCOM
007A52 4E722300       STOP #$2300
                  APU FIXD
007A56 13FC001E
       00050011  MOVE.B #FIXD,APUCOM
```

```
007A5E 4E722300        STOP #$2300
                     APURD A4,D2
007A62 3E3C0003        MOVE.W #$3,D7
007A66 E19A   @252 ROL.L #$8,D2
007A68 1414    MOVE.B (A4),D2
007A6A 51CFFFFA        DBRA D7,@252
007A6E 33C200040004 DA2CHFT2 MOVE.W D2,DACH2
                     TSTIMR A3,A1
007A74 0C390000
     00010035 @253 CMP.B #0,TSR
007A7C 67F6    BEQ @253
007A7E 13FC0001
     00010035  MOVE.B #1,TSR
007A86 B2CB    CMP.W A3,A1
007A88 6F00FF7A        BLE DA2CHFT1
007A8C 4E75    RTS
                    *
                    *
                    *
                    *###################################################################
                    *                            END OF PROGRAM
                    *###################################################################
                    ********************************************************************
                    *
                    *
007A8E 4BF87AA2      FIN LEA FINST,A5
007A92 4DF87AB2        LEA FINEND,A6
                     TRP14 OUT1CR
007A96 1E3C00E3        MOVE.B #OUT1CR,D7
007A9A 4E4E    TRAP #14
                     TRP14 TUTOR
007A9C 1E3C00E4        MOVE.B #TUTOR,D7
007AA0 4E4E    TRAP #14
007AA2 54     FINST DC.W 'THATS ALL FOLKS'
007AB2 20     FINEND DC.W ' '
                    *
                    *
```

```
2687             END

 ****** TOTAL ERRORS  0--     0 -- TOTAL LINES 2685

 APPROX1680 UNUSED SYMBOL TABLE ENTRIES

 @003    0060B0 @010    006178 @016    0062A2 @017    0062B6 @018    0062CA
 @022    00630A @030    006488 @037    00653A @053    006768 @061    0067CC
 @062    0067DC @063    0067F2 @064    006802 @065    00680E @068    006832
 @071    00685A @072    00686A @073    006876 @076    00689A @080    0068D0
 @084    006904 @086    00691C @087    00692C @089    00694E @093    006982
 @095    00699A @096    0069AA @097    0069BC @098    0069CC @100    0069E4
 @101    0069F4 @104    006A18 @105    006A28 @107    006A40 @108    006A50
 @111    006A74 @112    006A80 @116    006AB2 @118    006ACE @120    006B08
 @123    006B2C @126    006B50 @128    006B64 @130    006B7C @133    006BA0
 @135    006BB8 @137    006BD2 @139    006BEA @146    006C7A @147    006CA4
 @148    006CB4 @152    006CE4 @153    006CF0 @156    006D88 @159    006DB0
 @163    006DE0 @165    006E2C @166    006E38 @169    006E5C @171    006E74
 @185    0070A6 @186    0070B2 @187    0070C6 @190    0070EA @192    007106
 @198    007158 @199    007176 @201    00718E @202    0071A2 @204    0071BA
 @206    0071D2 @209    0071F6 @213    007258 @216    00727C @243    0079B8
 @245    007A12 @249    007A42 @252    007A66 @253    007A74 ACOS    000006
 AD1     00622A AD2     006238 ADCH1   020001 ADCH2   020003 ADCH3   020005
 ADCISR  00621C ADCNVRT 0061D0 ADCONT  00616A ADCONV  006126 ADEND   00623E
 ADEXADD000070 ADFIN   006204 ADFINE  00621A ADREAD  02000C ADSMPL  00007A
 ADST    0061EA ADSTE   006202 ADWAIT  0061AC ADWNSMP 0001E8 APISR   006240
 APUCOM  050011 APUEACK 04000E APUEXAD 000070 APUOPER 050001 ASIN    000005
 ATAN    000007 BLANK   0072D4 C2DFSQ  00641E C2DFSQ1 006444 CAEN    00641C
 CALCFIM0068BC CALCFRL 00693A CALCT   0067EA CALCU   0069B8 CALCW   006754
 CAR     0063CA CARCNV  000010 CARDWN  000008 CAREND  000F00 CARENEW 000F00
 CARETR  0074B8 CARSNEW 000C00 CARSTRT 000C00 CARTR   0074DE CAST    0063EE
 CCNVL   006470 CCNVLA  006494 CDCRTC  0064CA CDCRTC1 0064E6 CDCRTC2 0064F0
 CHSD    000034 CHSF    000015 CHSS    000074 CNTR    01002D CNVLV   00608E
 CNVLV1  006096 CNVLV2  00609C CNVLV3  0060C4 COS     000003 COUNTER 007F48
 CPR     010025 CT1T2   0064A0 D6464A  006572 D6464B  006598 DA      0072E8
 DA2CH   00795E DA2CHFT 0079D0 DA2CHFT1 007A04 DA2CHFT2 007A6E DAC1   0079A4
 DAC2    0079B2 DAC3    0079B0 DACA    007996 DACH1   040000 DACH2   040004
 DACH3   040008 DADD    00002C DAFIX   007206 DAFIXB  00728C DAMAX   007162
 DAMAX1  007164 DAMAX2  00716C DAMEN   007304 DANORM  007172 DANORM1 00719C
 DAOPT   007316 DASIG   007982 DCRTC   007F06 DDIV    00002F DIAFT   0075BA
 DIAPS   0075E2 DMUL    00002E DMUU    000036 DS2DEL  00001B DSUB    00002D
 DVD6464006564 E1      006100 E1E     006124 ECEN    006294 ECG     00624A
 ECGCAR  00746C ECGCNV  000010 ECGDWN  000008 ECGEND  000C00 ECGENEW 000C00
 ECGERR  00638E ECGSNEW 000900 ECGSTRT 000900 ECGTR   007492 ECNVL   0062F2
 ECST    00626E EDC     00650E EDC1    006522 EDCFDIA 007F1E EDCFSYS 007F1A
 EDCTDIA007F18 EDCTSYS 007F16 EDFSQ   0062DE EDFSQU  0062D4 EMOVE   006296
 ENDTBL  0060E2 ENTR    007946 ENTRE   00795C EQRS    006314 EQRS1   006322
 EQRS2   006328 EQRS3   006340 EQRS4   006356 EQRS5   006364 EQRS6   006380
 ERROR1  000000 EXP     00000A FADD    000010 FDIV    000013 FFT     006A8A
 FFT0    006AAC FFT1    006C40 FFT2    006C44 FFT3    006C4E FFT3LP  006C20
 FFT4    006D2C FFT5    006D40 FFT6    006D80 FFTLP4  006D08 FFTLP5  006D02
 FFTM    006EB6 FFTME   006EC4 FFTO5   006D50 FFTOLP5 006C6E FIN     007A8E
 FINDMAX006394 FINDMX  00639A FINEND  007AB2 FINST   007AA2 FIXD    00001E
 FIXS    00001F FLCN1   006E18 FLCN2   006E1E FLCN3   006E80 FLCNE   006E92
 FLEDC   007094 FLEDC1  0070BE FLTD    00001C FLTS    00001D FMAX    0063A2
 FMUL    000012 FSUB    000011 GETNUMA 0000E2 GETNUMD 0000E1 GOFIN   00605C
 HAM     006E9C HAM46   EB851E HAM54   8A3D70 HAME    006EB4 HAMM    006ADE
 HAMM1   006B60 HEX2DEC 0000EC HLTIMR  000000 HOLD    02000E I 007F30
 INCHE   0000F7 J 007F32 L       007F2A LE      007F2C LE1     007F2E
 LINKIT  0000FD LN      000009 LNN     00000A LOG     000008 MAIN    006012
```

MEMCLR	006006	MEMEND	005FFE	MEMST	000900	MSG	0075F6	MSGE	007628
N	000400	NEWTBL	0060DE	NM1	0003FF	NOPAPU	000000	NTROPT	0073EA
NTROPT1	007468	NV2	000200	OPT1	00762A	OPT1E	00765E	OPT2	007660
OPT2E	00769A	OPT3	00769C	OPT3E	0076D6	OPT4	0076D8	OPT4E	007718
OPT5	00771A	OPT5E	00775C	OPT6	00775E	OPT6E	00779C	OPT7	00779E
OPT7E	0077E0	OPT8	0077E2	OPT8E	007826	OPT9	007828	OPT9E	007862
OPTA	007864	OPTAE	007898	OPTB	00789A	OPTBE	0078D4	OPTC	0078D6
OPTCE	00790E	OPTF	007910	OPTFE	007944	OUT1CR	0000E3	OUTCH	0000F8
OUTPUT	0000F3	PCDEN	006644	PCDST	006628	PCEEN	00665E	PCEN	00660A
PCEST	006646	PCGCNV	000020	PCGCTR	00752A	PCGDIA	007578	PCGDWN	000002
PCGEND	001B00	PCGENDE	007F10	PCGENEW	001B00	PCGENSE	007F0E	PCGENST	002300
PCGETR	007508	PCGF	006F02	PCGS1A	007F08	PCGS1B	007F0A	PCGS2	007F0C
PCGSNEW	000F00	PCGSTRT	000F00	PCGSYS	007550	PCGT	0065A0	PCSEN	006626
PCSST	00660C	PCST	0065EE	PFDFT	006F90	PFDFTE	006FA2	PFEDC	007062
PFEDC1	00708A	PFEEN	006FBC	PFEN	006F7A	PFEST	006FA4	PFMDIA	007022
PFMDIA1	00703C	PFMDIA2	007044	PFMDIA3	007054	PFMSYS	006FBE	PFMSYS1	006FE6
PFMSYS2	006FEE	PFMSYS3	006FFE	PFPSYS	00700C	PFPSYS1	007018	PFSFT	006F7C
PFSFTE	006F8E	PFST	006F58	PNT2HX	0000E9	PNT4HX	0000E8	PNT6HX	0000E7
PNT8HX	0000E6	POPD	000038	POPF	000018	POPS	000078	PS	006EC6
PSE	006EDE	PSTOFT	007252	PTCD1	0066F4	PTCNVD	0066DC	PTCNVS	0066AE
PTCS1	0066D2	PTEDC	0066FE	PTEDC1	00674A	PTOD	000037	PTOF	000017
PTOS	000077	PTSQ1	00669C	PTSQU	006660	PUPI	00001A	PUTHEX	0000EA
PWR	00000B	QRS1	007F00	QRS2	007F02	QUIT	006064	QUITE	00608C
QUOT	007F44	SADD	00006C	SDIV	00006F	SEQNUM	007F12	SIN	000002
SMPS	006EE0	SMPSE	006F00	SMUL	00006E	SMUU	000076	SPAC	0072E4
SPCE	0072E6	SQRT	000001	SSUB	00006D	START	0000E5	STRTIMR	000001
SYSFT	0075A6	SYSPS	0075CE	T1TOT2	000098	T2PEAK	007F04	TAN	000004
TCR	010021	TMPENB	005EFC	TMPSTA	003700	TMPSTB	005700	TMPSTC	004700
TRCAREN	002300	TRCARST	002100	TRECGCHK	0020F0	TRECGEN	0020FC	TRECGST	001B00
TSR	010035	TUTOR	0000E4	TWOCOMA	0063C2	TWOCOMP	0063AC	UI	007F40
UR	007F3C	USRFNCS	0060CE	UTRER1	0060E6	VDR	007F00	WI	007F38
WR	007F34	XCHD	000039	XCHF	000019	XCHS	000079	XIEN	005700
XIST	004700	XREN	004700	XRENH	003EFC	XRST	003700		