NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

COMPUTERIZED CARDIAC MAPPING SYSTEM

by

THUTHUY BACH

A thesis submitted to
school of Graduate Studies and Research
in partial fulfillment of the
requirements for the degree of
Master in Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa

December 1990

ISBN 0-315-68024-5

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

I hereby declare that I am the sole author of this document. I authorize the University of Ottawa to lend this document to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Ottawa to reproduce this document by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

ABSTRACT

This thesis presents the results of the work on a computerized cardiac mapping system for recording simultaneously the signals from three electrodes placed at different sites on the heart in order to diagnose the sites causing arrhythmias. The signals are buffered, amplified, filtered and transferred to a personal computer via a multiplexing A/D converter. The computer controls the data acquisition process and stores the data in the memory. Activation times of the heart are estimated by a computer algorithm and displayed on a screen. The digitized waveforms and the vertical time marker of the activation time on each waveform are also displayed for verification. Corrections of computer-estimated activation times are possible using an interactive computer program. With this system, the region initiating an arrhythmia can be localized within a few minutes. The system has been used clinically and proved to be effective in detection of the sites of early activation causing cardiac arrhythmia.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my co-supervisors, Dr. S.Stuchly and Dr. F. Johnson for their guidance and encouragement throughout this work.

I would like to thank the other members of the Electrical Department, other members of The Ottawa Heart Institute and Dr. A.S.L. Tang for providing patients' data and a general advice. I also would like to thank Mr. Ronald Gagnon at Bell Northern Research for his great support during realization of this work.

The financial support of the Natural Science and Engineering Research Council of Canada is also gratefully acknowledged. Finally, a heartfelt thanks is extended to all the graduate students from whom I received valuable information.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation:

Ventricular arrhythmia is a disturbance in the normal sequence of the electrical process of the heart that can be both life-threatening and unresponsive to drug treatment [1, 2]. For these reasons, there has been an increased interest, in recent years, in the surgical treatment of this electrophysiological disorder. The purpose of the surgical approach to ventricular arrhythmia is the identification and ablation of the sites causing the arrhythmia [3]. The offending site might be either an area of slowed and fragmented conduction or an ectopic ventricular focus competing with the sinoatrial node, the normal cardiac pacemaker [4, 5]. The principle of ablation is to destroy the cells at this site in order to increase the resistance so that the arrhythmias cannot be sustained [6]. Determination of the site to be ablated requires a fast and reliable method for obtaining the activation time from the orderly waveforms recorded at various sites on the ventricular myocardium during arrhythmia.

1.2 Present state of knowledge:

The activation at a site on the heart refers to the occurrence of an electrical pulse on the electrical cardiogram (ECG). The time taken for the pulse to propagate from a reference point to this site is defined as the activation time [7]. A cardiac mapping system measures the potentials recorded from electrodes at various sites on the patient's heart. At each site , the activation is detected and the activation time is measured relative to the activation at a reference site [8.] The activation times are displayed for visual inspection. With this information, the surgeons can localize the sites having early or late activation times [9].

1.2.1 Data acquisition:

Many cardiac mapping systems have been developed. There are two methods that are currently used to collect electrical signals from the heart surface: single roving electrode [10] and multiple electrodes [11].

Single roving electrode:

The single roving electrode method was used at the Duke University Medical center to map the sequence of epicardial activation during sinus rhythm in patients with Wolff-Parkinson-White syndrome [10]. In this method, a single, hand-held probe, with an electrode at its tip, is positioned on the surface of the ventricles and the ECG is recorded from that area . The surgeon moves the electrode sequentially to a series of regions over the entire ventricular surfaces so that enough data is gathered at each region.

A reference electrode is held at a fixed site which is paced with a certain frequency. The fixed site can be on the atrium or the ventricle. The roving electrode measures the potential at different sites on the surface of the left or right ventricle during different cardiac cycles [12].

The advantage of this technique is that it does not require much hardware development since only one channel is processed at a time. However, this method is time consuming because the data is recorded sequentially, one site after the other. Furthermore, the ventricular arrhythmias usually cannot be sustained for the time required to record the data sequentially[13]. Finally, it is possible to have a variation in the activation sequence from beat to beat [14]. Therefore, this method is not always practical.

Multiple-electrodes:

The multiple-electrodes method is used at many research centers. One of these, the Department of Medicine and Pathology, Duke University [15], has developed a nylon mesh sock which is pulled over the heart. Fifty-two electrodes are attached to the sock and permit simultaneous recording of potentials from many sites on the ventricular epicardium during one cardiac cycle. One of the electrode on the sock is paced and used as the reference point for calculation of the local activation time at the other electrodes. This method allows rapid recording and analysis of epicardial electrical potentials and should meet the time constraints imposed during the study of ventricular arrhythmias in patients.

At the Washington University School of Medicine, a band of sixteen bipolar electrodes was developed [16]. This band permits simultaneous recording of potentials around the atrioventricular groove. Only one heart beat was required to obtained the activation times at the sixteen locations. This decreased cardiac manipulation during mapping and removed the need for cardiopulmonary bypass. This approach decreases markedly the time required for mapping and permits accurate study of non-sustained arrhythmias.

One of the disadvantages of the multiple-electrodes method is the large amount of data to be processed simultaneously. This requires a large investment in hardware to acquire the data as well as a powerful computer to process it.

Another disadvantage is that the recording electrodes may not be placed precisely over one of the standard predetermined grid points on the ventricles. This occurs because the sock is stretched when it is pulled over the heart and, because of its elasticity, it expands and contracts during diastole and systole. The fit of the sock over the heart is never perfect and in some regions, particularly at the apex, the contact between the recording electrodes and the ventricle is sometimes poor [6]. The errors in location may distort the maps.

Finally, leakage current is a problem: a few mapping systems are now available commercially, but some of them are not suitable for clinical use because they do not meet the leakage requirement [17]. It is easy to meet the leakage requirement for each individual electrode and its associated amplifier, but it is difficult to meet this requirement for many electrodes and amplifiers in parallel [18]. The mapping system that is used to record the electrical induction of arrhythmias during surgery must meet extremely stringent leakage requirements to reduce to infinitesimal limits the chance that the mapping system will induce an arrhythmia. It is also crucial that a system be electrically isolated from the patient so that in the event of malfunction, it does not produce a current that could be harmful to the patient [19].

1.2.2 . Activation detection methods:

Several methods to detect the activation have been proposed. These are:

Analog detector:

In this method, the activations are detected with a hardware slope detector [18]. An activation is defined as any point having a potential slope greater than a predefined threshold. This method is very fast because the activation is detected in real time. Furthermore, it does not require any software development and is less expensive than current digital systems. However, it does not always give the true activation because the signal amplitude varies from site to site.

Digital detector:

The present trend is to carry out signal processing tasks on the discrete or sampled representation of the signal using digital computers [20]. This has led to the development of digital algorithms for detecting the activation.

* Simple method [7]:

This algorithm was used to detect the activation based on a negative slope threshold. For each unipolar electrode,the window of 50 ms in which the activation most likely occurs is first detected based on the negative slope threshold. The computer finds all the samples within this window in which the potential slope is greater than the predefined slope threshold. If several samples meet this criterion then the sample that has the largest negative slope is taken as an activation for the electrode.

The advantage of this method is its simplicity, and the small time required for analysis . However, it is not reliable. First, the method can be applied to certain waveforms only. With the waveform as shown in Fig. 1.1, this algorithm might detect the first negative slope region instead the second one as the activation. It would not work in some arrhythmias in which the slowly conducted and small activation amplitude can have a deflection of less than the specified slope threshold [21].

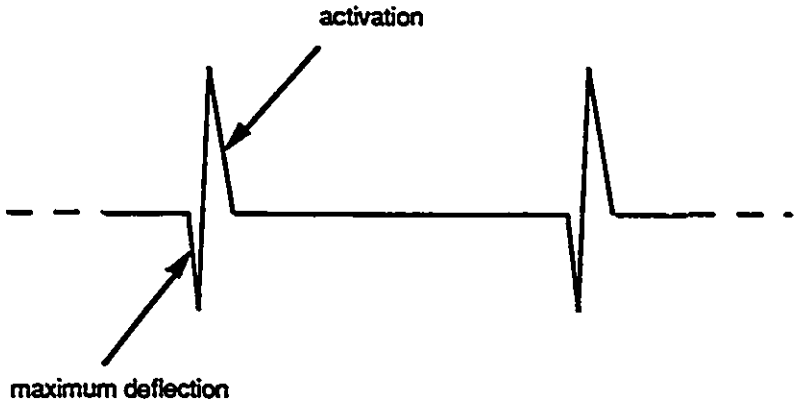

Figure 1.1 The maximum deflection region is not right at the acivation region.

* Interpolation technique:

This technique [20, 22] assumes that the recorded potentials are a continuous function pass through the sample points. The slope at every point is calculated based on the potential at (n -1) samples around it. For example with n=3 (Three-point Lagrange derivative):

$$f'(t_i) = [\ f(t_{i+1}) - f(t_{i-1})]\ /\ 2\ (\Delta t)$$

where Δt is sample time, $f(t_i)$ is a discrete sample value of the continuous electrogram and the activation is assumed to occur at time t_i, the time at which $f'(t_i)$ attained its maximum negative value.

This technique is more reliable than the simple method since it does not depend on any predetermined negative slope threshold. The disadvantage with this technique is the effect of noise; the negative slope at the activation may not be the maximum negative slope in the window. For example, with the waveform as shown in Fig. 1.1 the maximum negative slope at a point inside the activation region is much less than the negative slope at another point outside the activation region. Closer inspection revealed that the intrinsic deflection at the activation containing about 10 ms or 10 samples (with sampling rate of 1 kHz) of data between the maximum and minimum values as shown in Fig. 1.2 [20]. Therefore, this technique becomes more reliable when n is large because the probability of false detections due to noise is smaller. However, as n increases, the computation time also increases especially when the number of electrodes is large.

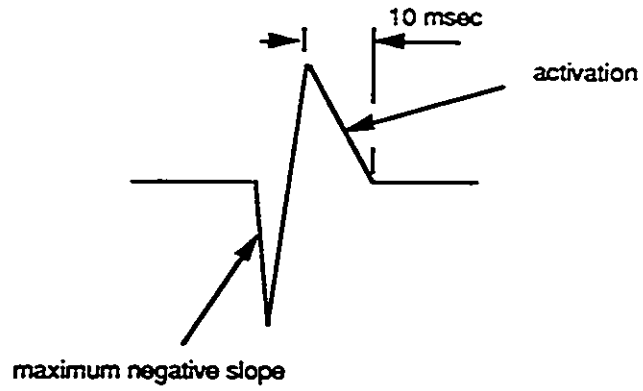

Figure 1.2 The intrinsic deflection at the acivation.

* Data-fitting technique:

"Data-fitting" is the term used in [20,22]. This technique is very similar to the interpolation technique except it assumes that the data contains noise of some form and the discrete series only gives an indication of the shape of the original signal. A data-fitting algorithm is used to approximate the original signal. Again, the location which has the maximum negative slope is considered as the activation. For examples:

Five-point second-order data-fitting technique (n=5):

$$f'(t_i) = 2f(t_{i+2}) + f(t_{i+1}) - f(t_{i-1}) - 2f(t_{i-2})$$

The n-point second-order data-fitting technique (n is any odd number):

$$f'(t_i) = \sum_{k=1}^{k=n/2} k[\, f(t_{i+k}) - f(t_{i-k}) \,]$$

The important factor determining the ease and accuracy of a derivative calculation at a particular point is the number of samples which must be included in the calculation. With n>11, the data values from an interval of more than 10 ms are used . Thus, in most cases, all of the data values used to calculate the derivative are located between the maximum and minimum values of the intrinsic deflection. However, as N increases, it is difficult to decide what type of function to fit to the data so no signal information is lost and the computation time also increases, especially when the number of electrodes is large.

Another problem arises when there are two equal negative deflections as shown in Fig. 1.3. Such a situation occurs when a single electrode detects the impulses in several adjacent myofibers. An electrode has finite physical dimensions, and many myofibers are adjacent to the electrode, but are separated from it by variable amounts of interstitial fluid and connective tissue. The electrode also records potentials from myofibers that are not adjacent to it [23]. Thus, a standard electrode does not always exhibit a single discrete activation but many fragmented activation fronts. Therefore, even with the high-order data-fitting technique, we would not be able to detect which activation front is caused by the myofibers in contact with the electrode unless some other criteria must be considered.

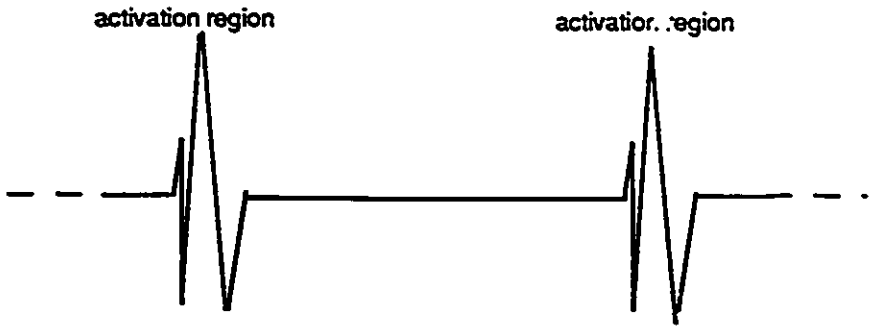

Figure 1.3 Two fragmented activation fronts occur in the same activation region.

1.3 Objectives:

The main objective of this project is to research, develop, and design a computerized cardiac mapping system which would be faster, more efficient, more reliable and more economical than the existing systems. The second objective is to test such a system in a clinical setting.

1.4 Organization of the thesis:

Chapter 2: This chapter describes, at the system level, the various functional units of the proposed cardiac mapping system. These are the data-acquisition, activation detection and activation-time mapping units. The various blocks in each unit are described.

Chapter 3: This chapter gives detailed description of various hardware blocks such as: selectors, buffers, filters, amplifiers and converters to condition the signals. The function and diagram of each unit are presented.

Chapter 4: This chapter provides detailed description of the proposed activation detection algorithm. The advantages and disadvantages of the method are discussed . Finally, the software implementation of this algorithm and the mapping unit are presented.

Chapter 5: This chapter presents simulation and clinical results obtained using the computerized cardiac mapping system. The various test waveforms used for simulation are presented. The proposed activation detection algorithm is compared with three other algorithms. Finally, a statistical analysis of the clinical results is presented.

Chapter 6: The efficiency, reliability, computation time and the cost of the proposed system are discussed and compared with other existing systems. Extensions of the studies in computerized cardiac mapping are also explored.

Appendix A is the listing of the program which are used to detect the activation.

Appendix B is the listing of the mapping program.

Appendix C is the listing of data-retrieval program.

Appendix D contains detailed hardware configuration.

Appendix E describes man-machine interface.

CHAPTER 2

DESCRIPTION OF THE SYSTEM

2.1 Introduction:

The computerized cardiac mapping system contains three major units as shown in Fig. 2.1:

- Data-acquisition unit.
- Activation detection unit.
- Activation-time mapping unit.

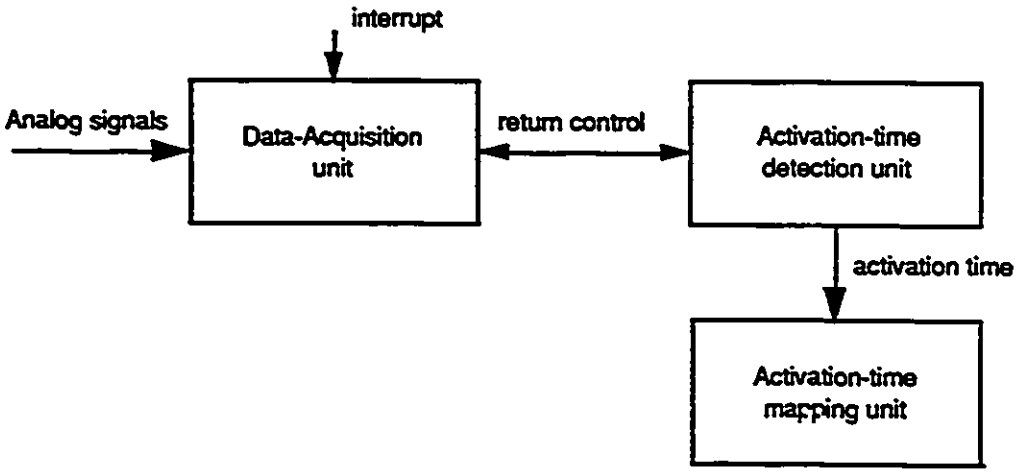

Figure 2.1 System block diagram.

Data-acquisition unit:

Signals from many electrodes may be recorded simultaneously. At the present time, only three channels are used but the system can be expanded to handle many more channels. The number of channels is limited by the processing speed of the computer and the conversion time of the A/D converters. Two fixed electrodes are generally sutured to both the atrium and the ventricle for purposes of recording and pacing with a frequency from 2 Hz to 5 Hz. Either electrode may serve a dual purpose in providing a reference to which the data are related: The atrial reference is used when the atrium is being paced and the ventricular reference is used when the ventricle is paced. This also affords a convenient method for monitoring the atrioventricular conduction time, and permits prompt recognition of the development of a junctional rhythm or block in the accessory pathway due to inadvertent trauma during mapping [24]. The site for the ventricular reference should preferably be devoid of epicardial fat and blood vessels and yet be located as close as possible to the atrioventricular ring in the area assumed to be the location of the accessory pathway [12]. The third channel is used to record the signal from the roving electrode positioned at various sites on the left or right ventricle . The position of these electrodes are shown in Fig. 2.2.

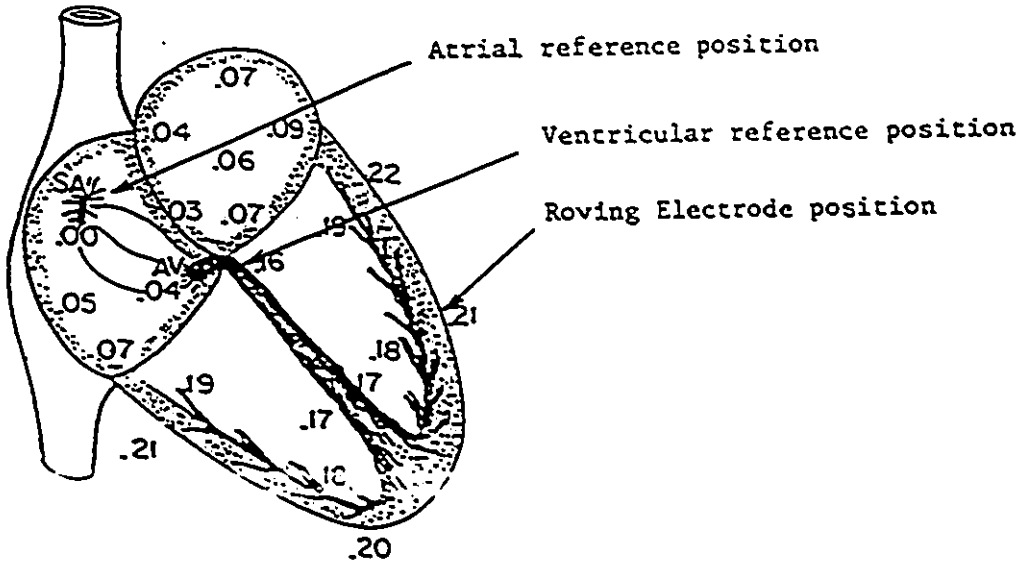

Figure 2.2 The position of the electrodes on the atrium and the ventricles.

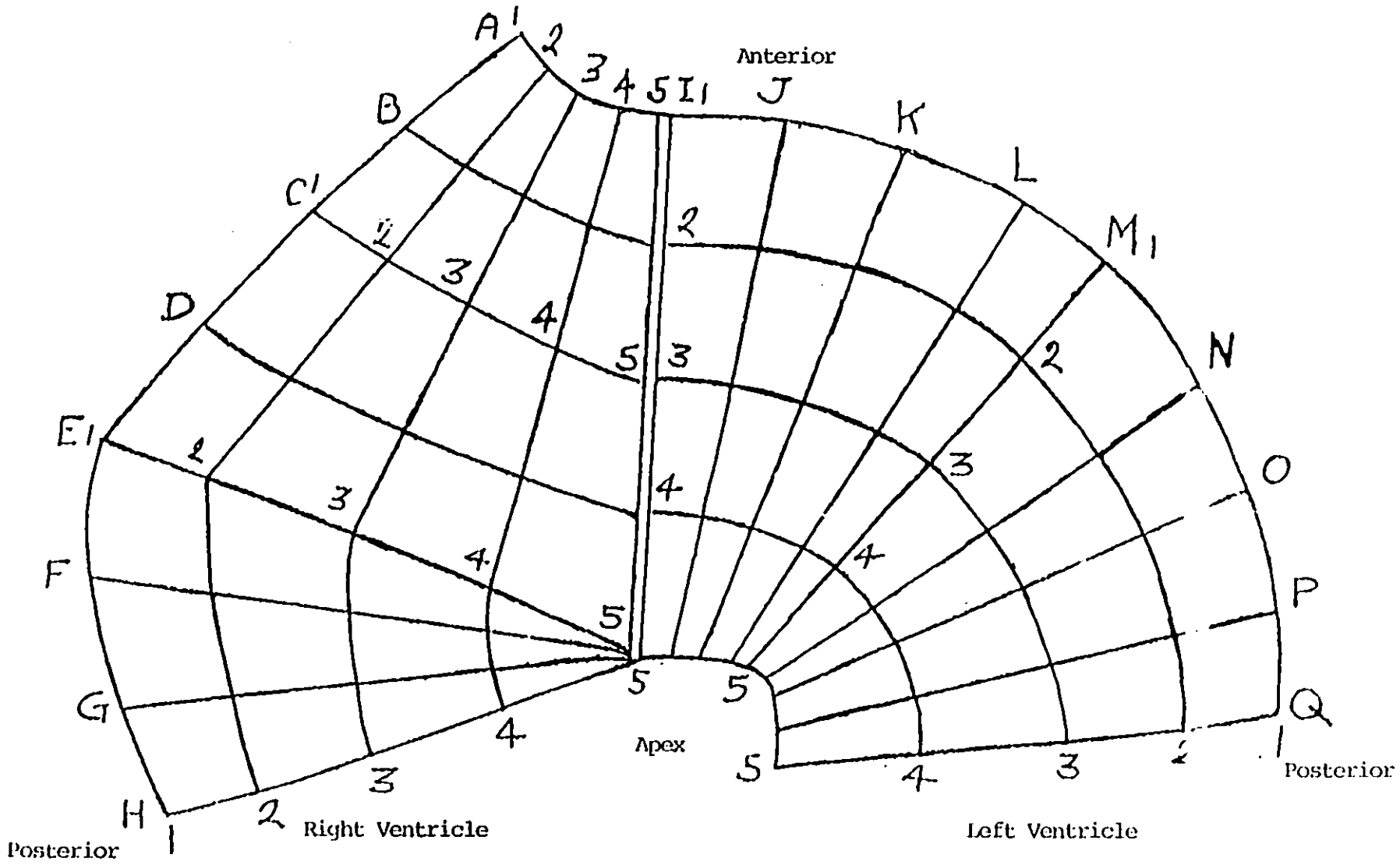

Figure 2.3 The ventricular map.

The total ventricular surface is subdivided into 17 regions labeled from A to Q as shown in Fig. 2.3. The outline of this map represents the ventricles separated at the posterior interventricular groove and spread flat. The measurements are made at the center of each region with the roving electrode [13].

Activation detection unit:

The activation detection algorithm first detects the activation at the reference site and then it detects the activation at a selected site on the ventricles. The activation time at the selected site is calculated as the difference between these two times.

Activation-time mapping unit:

The activation times are entered on a ventricular chart using different colors for different ranges of activation times. The waveform and the numerical value of the activation times are also displayed, in tabular form, for verification and modification.

2.2 The hardware of the system:

The system is designed to receive and condition the signals from various electrodes. The basic blocks of the hardware unit are shown in Fig. 2.4 . These included:

* Selectors; for selection of three channels from the six available channels.
* High-input impedance buffer amplifiers.
* RC low-pass filters and AC coupling.
* Variable gain amplifiers.
* Multiplexing A/D converter system with interface.
* A 286 personal computer with a VGA graphics terminal.

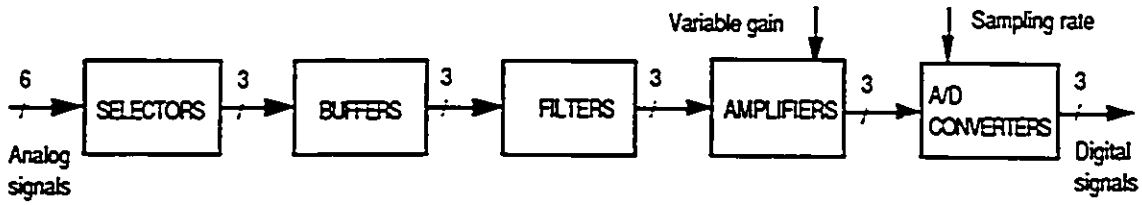

Figure 2.4 Basic blocks of the hardware.

In the system that have been developed at the University of Ottawa Heart Institute, there are six channels available to collect the signals from the electrodes. These signals are already electrically isolated before they are processed by this computerized cardiac mapping system and other equipment in the operating room . Only three of the six available channels are used during operations. The signals from the three selected channels are fed to a set of buffers (with input impedance of 1 MΩ) and a set of RC low-pass filters with AC coupling. Each channel of the system has a variable gain amplifier. The output signals from the amplifiers are fed to a multiplexing A/D converter system. The timing for the A/D conversion is supplied by a 5 MHz crystal oscillator and a series of counters to give a 1 kHz sampling rate, which has been shown to be the minimum sampling rate necessary for epicardial data recording [25]. The sampling rate can be modified by a program-controlled interface. An interrupt is generated every millisecond and the contents of the A/D converter are transferred to the buffers in the memory unit. For more details of hardware description, please refer to Chapter 3.

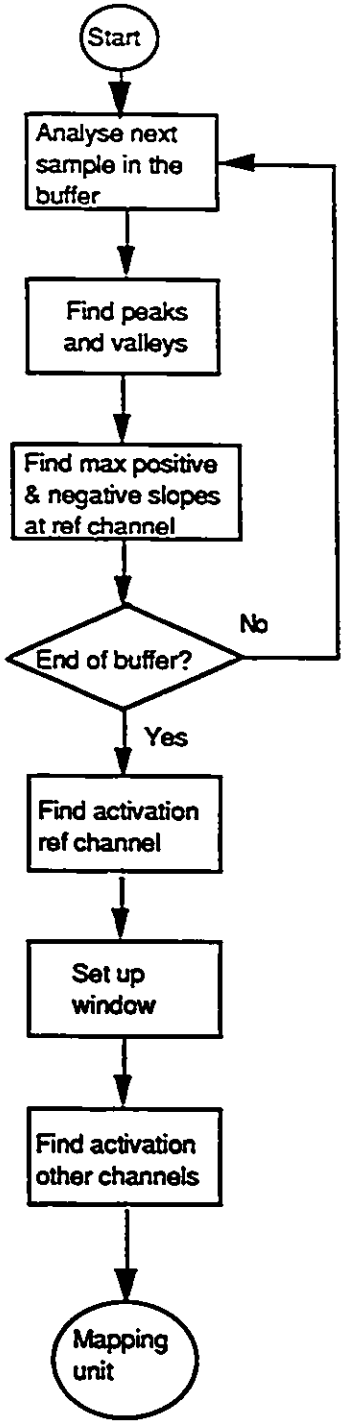

Figure 2.5 Flowchart of the activation detector.

2.3 Software of the system:

The activation detection and activation-time mapping units are done by software residing in the personal computer.

2.3.1 Activation detection unit:

The digital data from the A/D converters are transferred to a buffer in the memory unit. Each sample in the buffer is analysed to find the region of maximum amplitude, maximum positive and maximum negative slopes in order to determine the activation time. After all the samples in the buffer have been analysed, the activation time found at each channel is sent to the mapping unit for display. The flow chart of the activation detection unit is shown in Fig. 2.5.

* **Buffer:**

The reference point is paced with the frequency of 2 Hz to 5 Hz. If the heart paces by itself, its own pacing frequency would be above 1.2 Hz (one stimulus every 850 ms). In addition, the activation at the other channels usually happens within 350 ms after the activation of the reference channel. Therefore, we need at least 1200 samples of continuous data in order to determine the activation at all the channels as shown in Fig. 2.6.

* **Peaks and valleys:**

The maximum amplitude is one of the criteria used to detect the activation. Therefore, all the peaks and valleys are detected to find out which peak has the maximum amplitude. A sample point is a peak if its potential is greater than its two neighbors' potentials and it is a valley if its potential is less than its two neighbors' potentials:

if $f(t_i) > f(t_{i-1})$ and $f(t_i) > f(t_{i+1})$ then t_i is a peak.

if $f(t_i) < f(t_{i-1})$ and $f(t_i) < f(t_{i+1})$ then t_i is a valley.

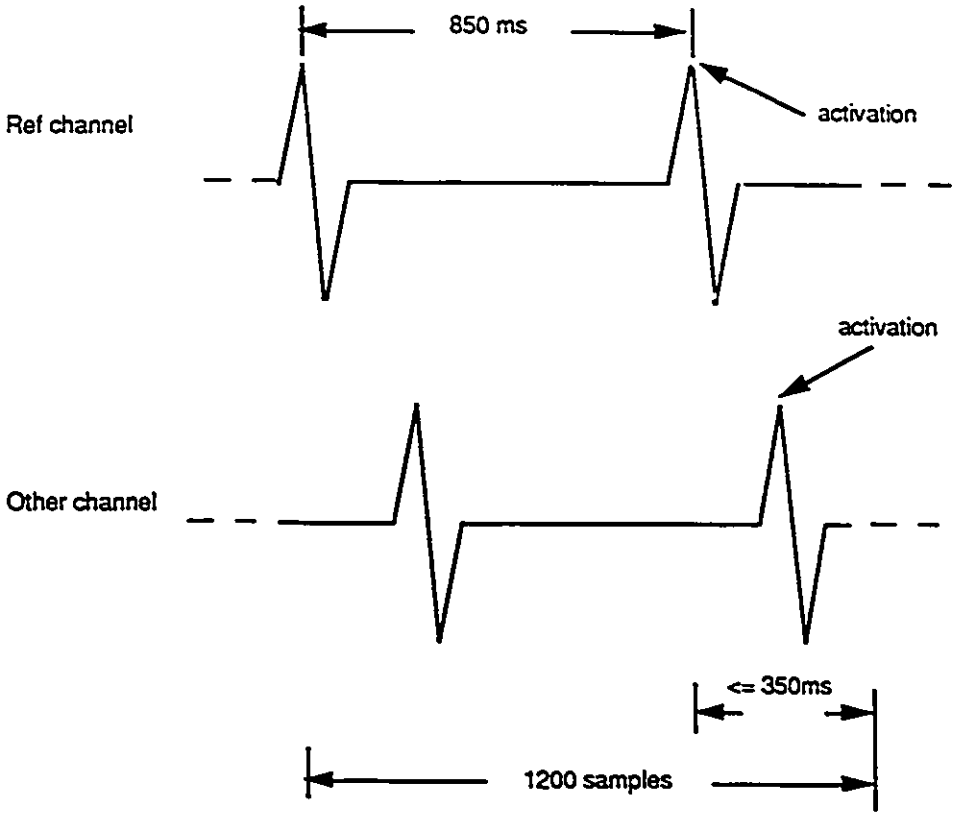

Figure 2.6 Buffer of continuous 1200 samples of data.

*** The positive and negative slopes:**

In addition to the maximum amplitude, the maximum positive and negative slopes are also used to detect the activation. Therefore, The average positive slope between a peak and a valley and the average negative slope between a peak and a valley are calculated to find out which regions have the largest maximum slopes.

*** The activation at the reference channel:**

The activation time at a site on the ventricles is measured relative to the activation at the reference channel. Therefore, the activation at the reference channel must be detected before searching for the activation at the other channels.

* Set up Window:

As mentioned earlier, the activation at other channels usually occurs within 350 ms of the activation of the reference channel. Therefore, instead of looking for the activation of the other channels in the whole segment of 1200 samples of data, we can search for the activation at these channels in the window in which they will most likely occur. The window of interest is shown in Fig. 2.7.

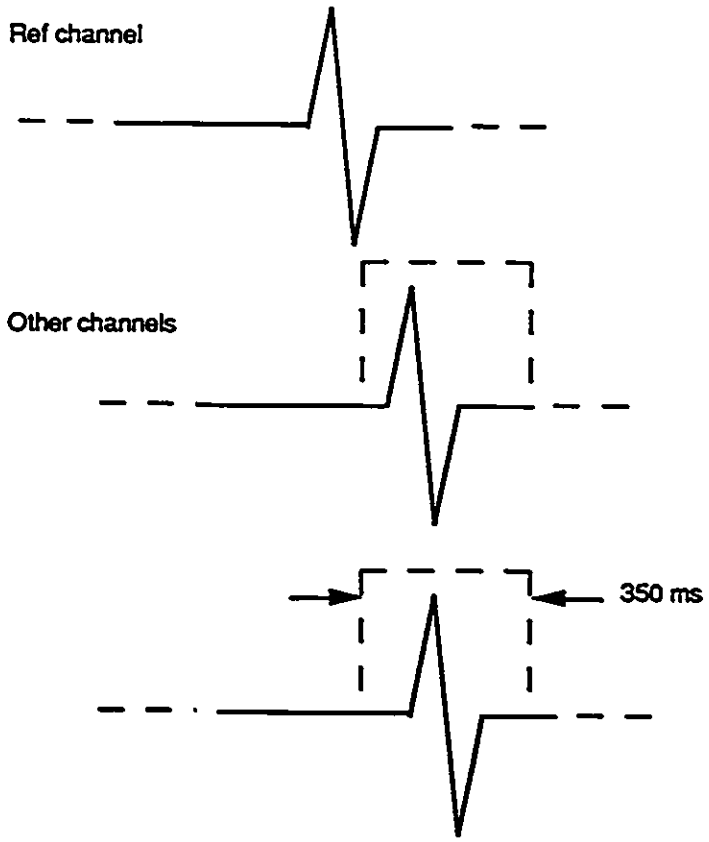

Figure 2.7 Window used for searching the activation time at the other channels.

* **The activation detection algorithm:**

The algorithm considers the activation as a point of maximum amplitude at which the transient between the region of maximum positive slope and the region of maximum negative slope takes place. Thus, it uses the maximum amplitude, maximum positive and maximum negative slopes as criteria to detect the activation.(Please refer to section 4.1 for more details about this algorithm).

2.3.2 Activation-time mapping unit:

The sub-units of the activation time mapping unit are shown in Fig. 2.8.

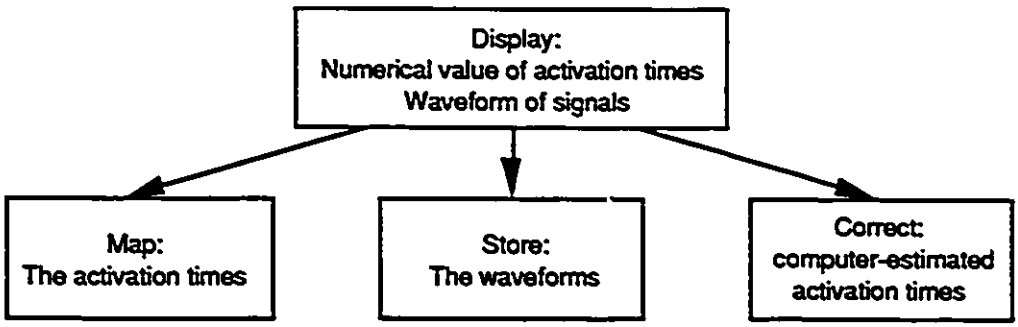

Figure 2.8 The sub-units of the activation time mapping unit.

- **Display:**

The numerical value of the activation time at each location are displayed on the screen. The waveform at each channel and the sample where the activation takes place is also indicated by a vertical cursor on the waveform.

- **Mapping:**

The activation time at each location on the ventricles is mapped with color code. Each color represents a certain range of activation time. The ranges of the activation time are predetermined based on the maximum and minimum activation time determined on the ventricles. At the present time, six colors are used to map the activation time of the ventricles. These colors and the corresponding range of the activation time are shown in Table 2.1. Hence, the region which is mapped with red color is the one that has early activation and the one which is mapped with blue color is the one that has late activation.

Color	Activation Time
Red	<= min + scale
Pink	<= min + 2*scale
Yellow	<= min + 3*scale
White	<= min + 4*scale
Green	<= min + 5*scale
Blue	> min + 5*scale

Table 2.1 The corresponding color for the activation time
(scale = (max-min) / 6)

* **Storage:**

These waveforms can be stored into the hard disk for later study.

* **Modification:**

The point of activation, as determined by the computer, is indicated by the vertical cursor on the waveform. Misinterpretations of the computer may be corrected if necessary: The vertical cursor can be moved along the signal to the point of activation selected by the user.

CHAPTER 3

HARDWARE

This chapter describes the circuit diagram of each hardware sub-unit of the system. For more details about hardware components, please refer to Appendix D.

3.1 Selectors:

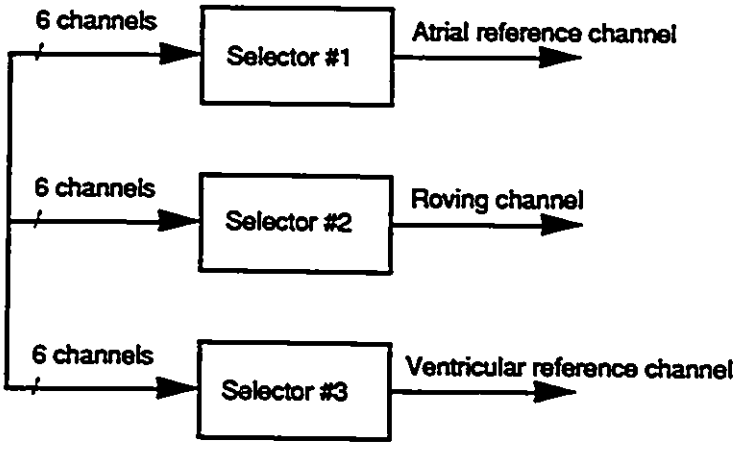

Figure 3.1 Selector configuration.

The selectors select any three of the six available channels as shown in Fig. 3.1 to record the data at the atrial reference, the ventricular reference and the roving electrodes.

3.2 Buffers:

Fig. 3.2 shows the circuit diagram of the buffer amplifier in each channel. The amplifier has a high input impedance and low input capacitance in order to prevent distortion of the input signal. Two 1 kΩ resistors are in series with op-amps inputs to limit the currents in case of malfunction. Furthermore, the signal ground and the computer ground are not directly connected together to provide rejection of common mode noise[26].

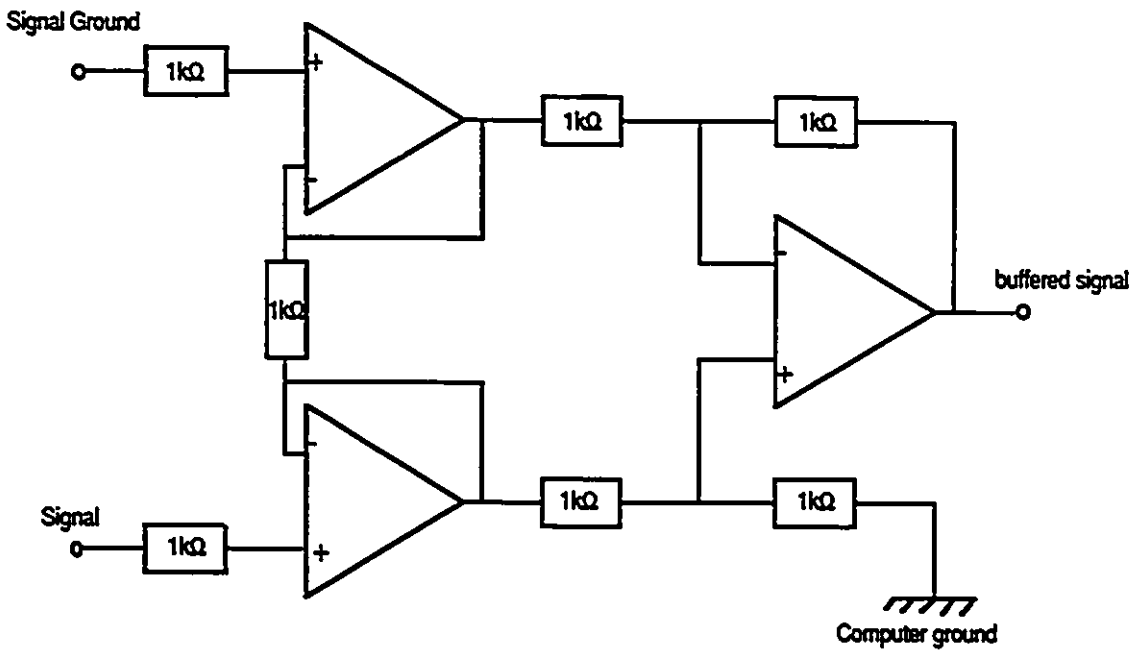

Figure 3.2 Buffer Configuration.

3.3 Filters:

Figures 3.3 and 3.4 show the filtering circuit diagram at the reference channel and the other channels, respectively.

An AC coupling is used to remove any DC component. This RC stage (100 kΩ and 3 μF) has a cut-off frequency of 0.5 Hz.

A low-pass filter, implemented with one stage RC filter (3.24 kΩ and 0.1 μF), is used to prevent aliasing during digitization and to reduce high frequency noise. The cut-off frequency (500 Hz) is chosen as half of the sampling frequency (1 kHz). This was determined to be adequate, although a second order filter may be needed so that the cut-off frequency is less than the Nyquist frequency. The frequency response is plotted in Fig. 3.5.

3.4 Amplifiers:

The analog to digital converter (A/D converter) operates in the range from -5 volts to +5 volts. The amplitude of the input signal can vary significantly depending on:

- Recording mode (unipolar or bipolar).
- Method of measurement (epicardially or endocardially).
- Tissue where the electrode is placed (ischemic or normal).

The amplifier is used to amplify the signal to the optimum level that is suitable for the A/D converter as well as for detection of the activation. The circuit diagram of the amplifier in the reference channel and those in the other channels are shown in Figures 3.3 and 3.4, respectively. The output from the isolated preamplifiers (not shown) were of the order of 1 volt.

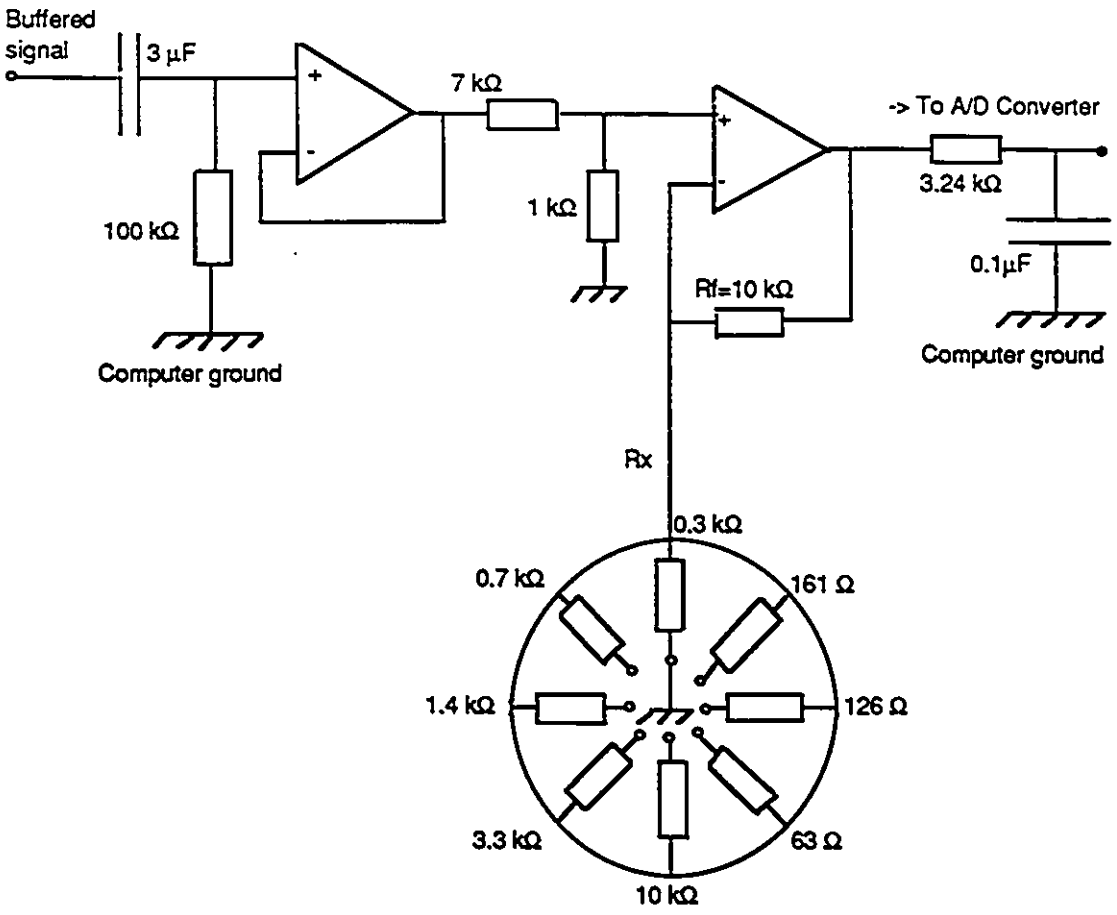

Figure 3.3 Filter and amplifier circuit diagram in the reference channel.

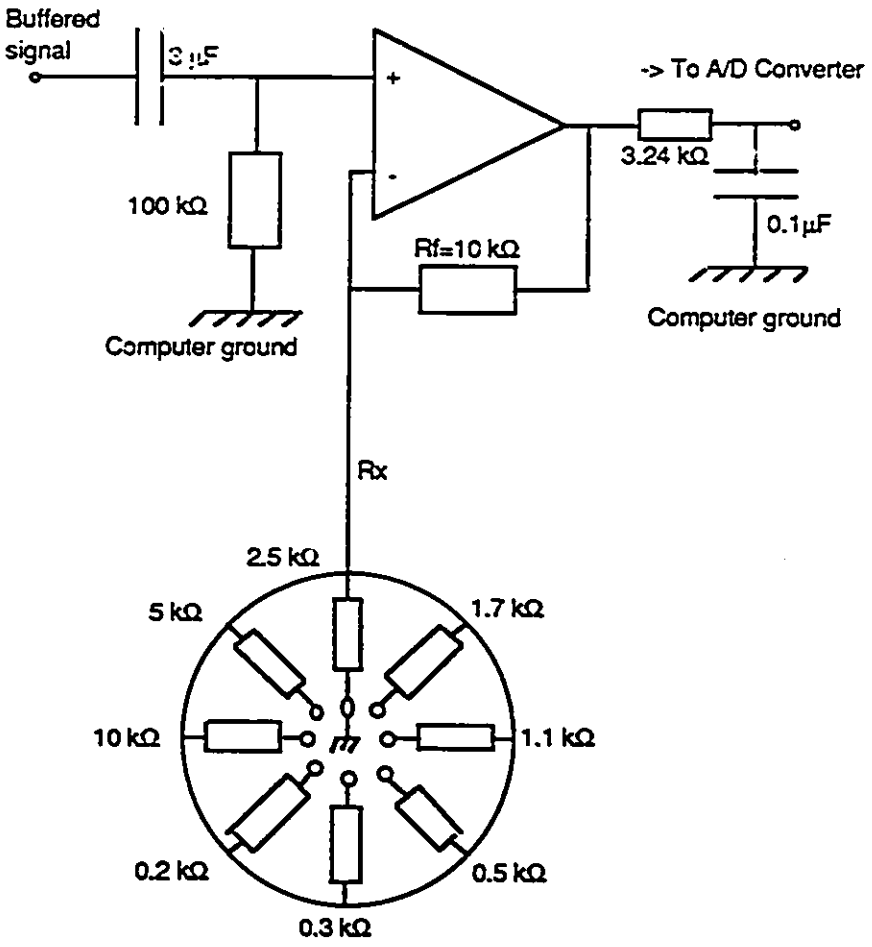

Figure 3.4 Filter and amplifier circuit diagram in the others channels (non-pacing channels).

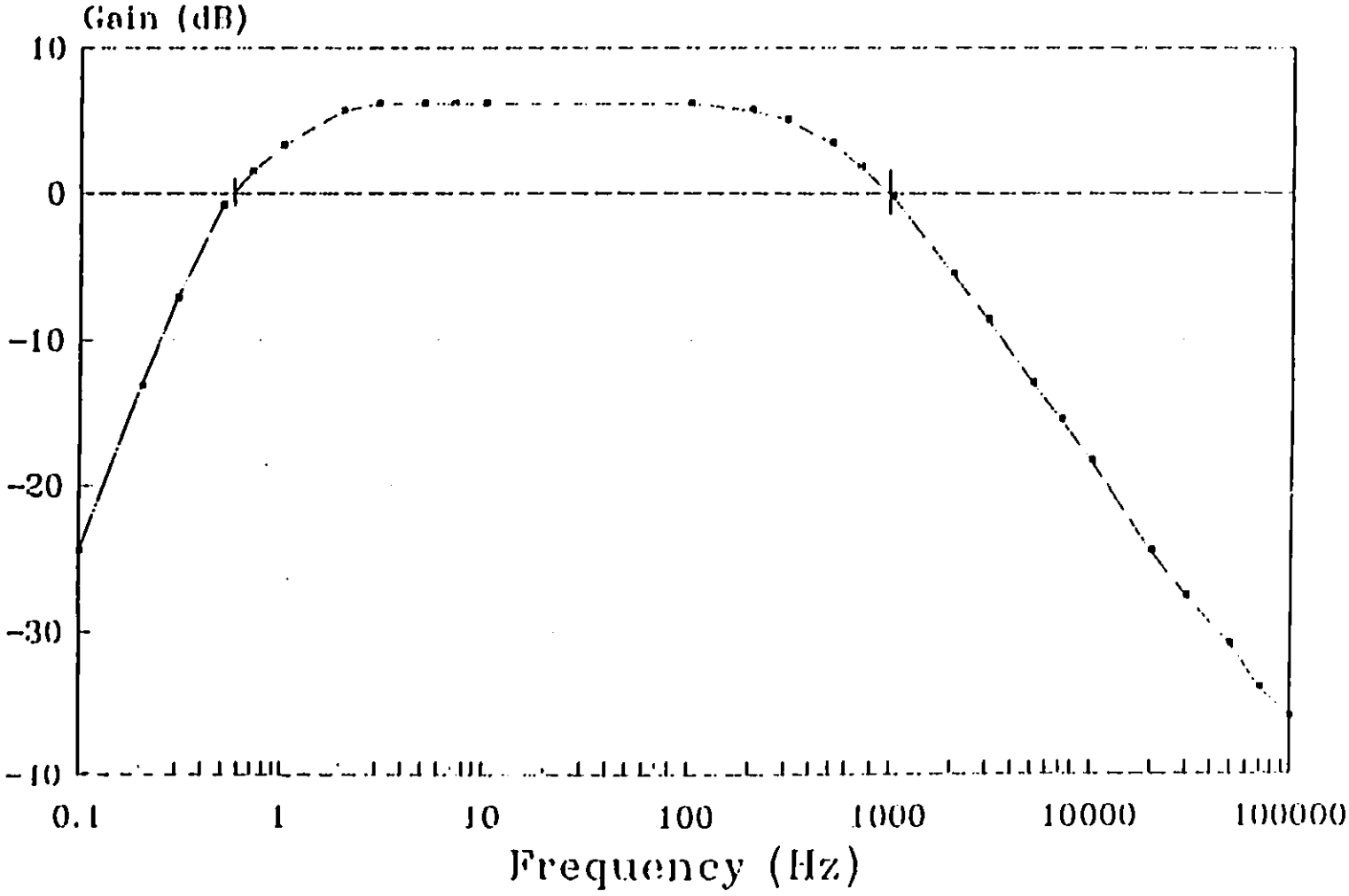

Figure 3.5 Gain vs frequency

In the fixed gain amplification mode, the gain for all the amplifiers is kept constant. Recording in this mode is simpler than recording at variable gain because the hardware to change the gains setting of the amplifiers does not have to be developed. Furthermore, it is easier to differentiate which signals have large or small amplitudes. The signals are in the range from 1 mV to 200 mV. However, the signals created by the pacing stimulus can exceed 500 mV. A 12-bit A/D converter, with the range of +/- 5 volts , has a step size of 2.4 mV. This step size cannot adequately characterize any signal below 24 mV. Therefore, a wide and independent gain range is needed for each channel in order to adequately characterize all input signals.

Variable gain amplification refers to the ability to change the gain of each amplifier individually and independently. As mentioned earlier, the amplitude of the signals depends on the types of electrode, the location where the recording is made and the the tissue where the electrode is applied to. For example:

* For unipolar recording, the signals are normally in the range from 20 mV to 60 mV on the ventricles and from 5 mV to 15 mV on the atrium [12]. For bipolar recording , the signals are normally greater than 3 mV [27].

* The pacing stimulus at the reference site usually has an amplitude of greater than 500 mV. However, the amplitude of the signals recorded at the other sites is not greater than 70 mV [21].

* The amplitude of the signals recorded at the ischemic tissue is usually much smaller than the amplitude of the signals recorded at the normal tissue [21].

Therefore, the variable gain mode is used. In this mode, the output signals have the same amplitude. Since the signals are optimally amplified to match the dynamic range of the A/D converter, the signal to noise ratio (S/N) is better. The connections between the amplifier and A/D converter must be shielded to maintain an acceptable S/N of 60dB.

* The reference channel:

The filtering and amplifying circuit for the reference channel is shown in Fig. 3.3. It is very similar to the filtering and amplifying circuit of the other channels. The only difference is that the signals are buffered and attenuated by 1/8. Since the reference electrode is placed where the pacing stimulus is applied and the amplitude of the stimulus is usually larger than 500 mV. Without the attenuation, the amplifiers and A/D converter may saturate. The gain at this channel can also be changed manually . The total gain in this channel depends not only on the gain at the amplifier but also on the attenuator setting. The gains and the corresponding Rx at this channel is shown in Table 3.2:

* **The roving channel and the other channel:**

The gain of the op-amps can be changed using different Rx:

Gain = 1 + Rf/Rx

The variable gains and the corresponding Rx in the roving channel and the other channels are presented in Tables 3.2 and 3.3, respectively:

Rx (Ω)	Gain=1+Rf/Rx at the amplifier	Total gain
10 k	2.0	0.2
3.3 k	4.0	0.5
1.4 k	8.1	1.0
0.7 k	15.3	1.9
0.3 k	34.3	4.3
161	63.1	7.9
126	80.4	10.0
63	159.7	20.0

Table 3.1 Gain and coressponding Rx (Rf=10 kΩ) in the reference channel (attenuation=1/8)

Rx (kΩ)	Gain=1+Rf/Rx
10	2.0
5	3.0
2.5	5.0
1.7	6.9
1.1	10.1
0.5	21.0
0.3	34.3
0.2	51.0

Table 3.2 Gain versus Rx (Rf=10 kΩ) in the other channels.

3.5 A/D converters:

The A/D converters on AD1000 card are manufactured by Real Time Devices Inc. [28]. The card is placed in a slot of the personal computer. Since the computer and the terminal are manufactured for commercial use, they have relative large leakage currents (520 μA when the computer power switch was off and more than 1000 μA when it was on). These figures made them unsuitable for clinical use. Therefore, an isolating transformer (Hammond 171E) was used to reduce the system ground leakage current below 100μA.

The A/D converter converts the input analog signal to a digital signal for processing , storage and analysis. A sampling rate of 1000 samples/s is used because recordings from the surface of the ventricles with unipolar electrodes and with bipolar electrodes whose interpolar spacing is at least 1 mm can be adequately characterized with the sampling rate of 1000 samples/s [18]. The use of a smaller sampling rate may result in degradation of the waveform in a manner that can lead to errors in activation time measurements and misinterpretation of the wave shape. A higher sampling rate might be required if detailed features of the waveforms are of interest. However, these detailed features are not necessary in the present study. Furthermore, the increasing cost of data storage is associated with increasing sampling rates especially when the number of channels is large [25].

CHAPTER 4

SOFTWARE

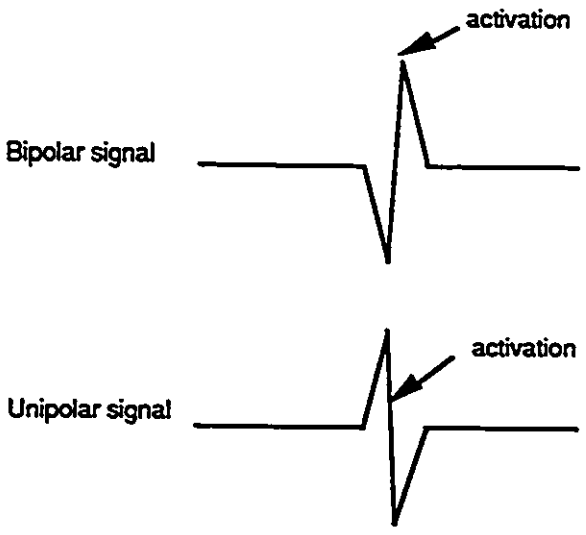

Figure 4.1 Determination of local activation at bipolar and unipolar electrograms.

4.1 The proposed algorithm:

The activation time at a unipolar electrode site is taken as the point of the maximum negative slope, and the activation time at a bipolar electrode site is where the maximum amplitude occurs [29]. However, because of the noise, measurements of the activation time from the bipolar signals cannot be based only on the maximum amplitude. It is more reliably based on the time of occurrence of maximum absolute slope [23, 30, 31]. Thus, an algorithm is proposed which uses both the maximum positive and negative slopes to detect activation times from either unipolar or bipolar signals. Closer inspection reveals that for either unipolar or bipolar signals, the region of activation usually contains a point of maximum amplitude at which the transient between the region of maximum positive slope and the region of maximum negative slope takes place. Therefore, this point is taken as the time of activation.

This suggestion would not violate what have been used so far in [23, 30, 31] because for the bipolar signal, the point of maximum amplitude is truly the point of activation. For a unipolar signal, the difference between the point of the maximum negative slope and the point of the maximum amplitude is less than 2 ms. The sampling rate of 1000 samples/s may cause an error of up to 1 ms. Furthermore, the local activation time at any point is measured relative to the activation of the reference channel, and if the errors are random then the rms error = $\sqrt{2^2 + 1^2 + 2^2 + 1^2}$ = 3.3 ms. The local activation time at any point on the ventricles is usually larger than 200 ms. Thus, an error of less than 1.65 % is acceptable for the estimation of pathway pathology.

The algorithm contains two major sections: Data-extraction and data-justification.

(1) Data-extraction: Two types of features are extracted: local amplitude extrema and local slope extrema.

- Amplitude extrema (peaks and valleys) are located by searching for a change in the sign of the first derivative as follows:

If $f(t_i) > f(t_{i-1})$ and $f(t_i) > f(t_{i+1})$ then i is a peak.

If $f(t_i) < f(t_{i-1})$ and $f(t_i) < f(t_{i+1})$ then i is considered as a valley.

- Slope extrema (the sides of the peaks and valleys) are calculated as follows:
  Positive slope is the average slope between a valley and a peak.
  Negative slope is the average slope between a peak and a valley.

The extrema of all the regions are compared to find the regions of maximum amplitude, maximum positive and maximum negative slopes as shown in Fig. 4.2.

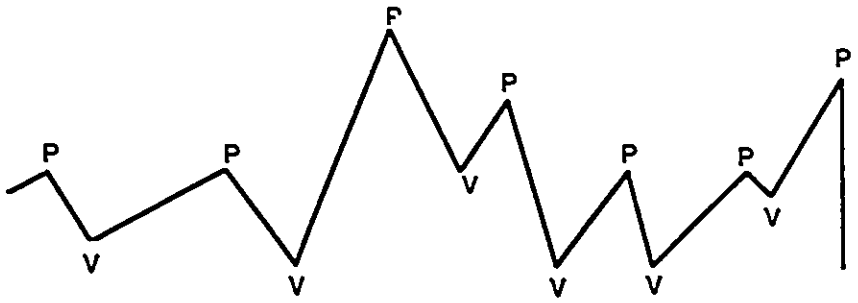

Figure 4.2 The amplitude extrema (peaks and valleys) of the signals.

(2) Data-justification:

The time of activation is taken to be where the peak of the maximum amplitude, maximum positive and maximum negative slopes occur. If these regions do not coincide then their amplitudes and the positive and negative slopes are compared to determine which region is truly the region of activation and the peak in this region is taken as the point of activation.

The method is illustrated below:

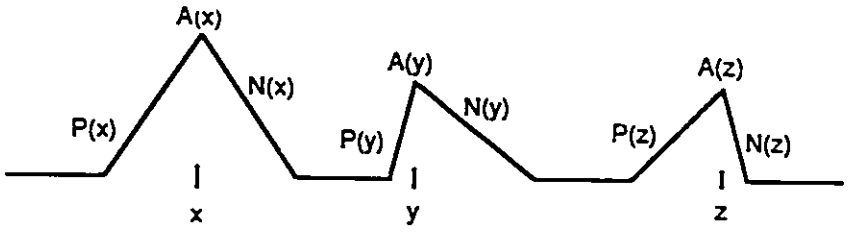

Figure 4.3 One example: when the regions of maximum amplitude, maximum average positive and maximum average negative slopes are not coincident.

- x is the region of having maximum amplitude A(x).
- y is the region of having maximum average positive slope P(y).
- z is the region of having maximum average negative slope N(z).

where A(i), P(i) and N(i) are the amplitude, average positive slope and average negative slope of the region i, respectively as shown in Fig. 4.3. These factors are converted into a common factor called R(i):

$$R(x) = 1 + P(x)/P(y) + N(x)/N(z)$$
$$R(y) = A(y)/A(x) + 1 + N(y)/N(z)$$
$$R(z) = A(z)/A(x) + P(z)/P(y) + 1$$

The region that has maximum common factor is considered as the region of activation :

$$R(\text{ activation }) = \text{Max } \{ R(x), R(y), R(z) \}$$

These are several advantages of the proposed algorithm:

Although the algorithm appears complicated, it is more reliable, has better noise immunity and is faster. It is reliable since it does not depend on only the maximum amplitude or maximum slopes but on all of these criteria to detect the time of activation.

The results are not affected by noise or noise spikes because the slope is not calculated at a specific point based on the the potential of the number of points around but the average slope in each region of upslope or downslope.

It requires little computation time since it does not calculate the slope at every single point in the window. The flow chart of this algorithm is shown in Fig. 4.4.

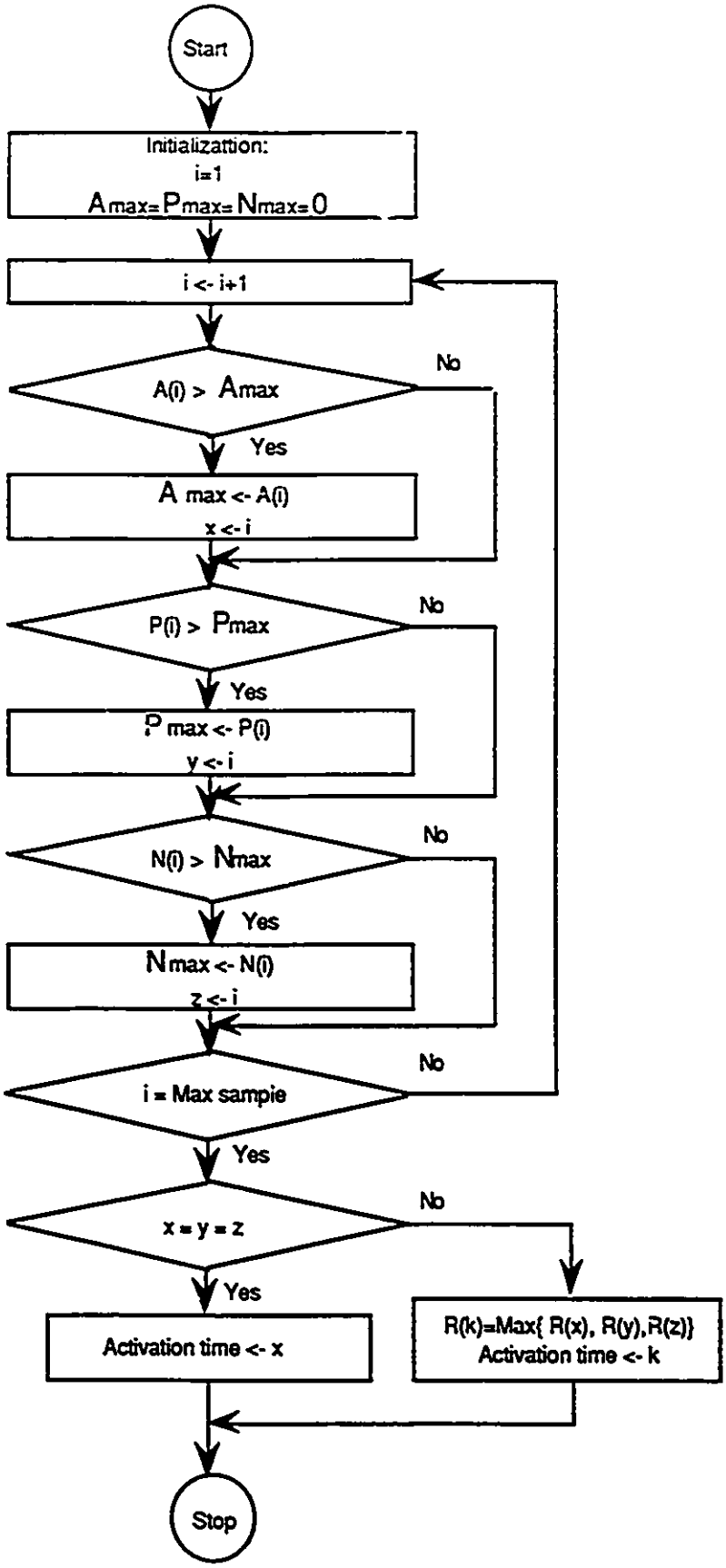

Figure 4.4 Flowchart of the proposed algorithm.

4.2 Software implementation:

* Activation-time detection module:

This module controls the data acquisition and detects the activation time at each channel. The two processes run concurrently.

The data-acquisition is interrupt driven. The interrupt handler is first enabled by a data-acquisition request and an interrupt is generated at every millisecond until the buffer is full.

Each interrupt initiates the A/D conversion and transfers the data into a buffer. Control is then returned to the second process to continue analysis of the data until the next interrupt. The flow chart of the data-acquisition control process is shown in Fig. 4.5 and the flow chart of the data-analysis process is shown in Fig. 2.5. The program listing of this module is presented in Appendix A.

Since the first process takes less than 0.1 ms to record the digital data at all the channels, we can make use of more than 0.9 ms to analyse the data during each sampling period. After a buffer of 1200 samples has been collected, the activation time at each channel is found and sent to the mapping unit for display. If any other request is issued, the interrupt handler is enabled again and the new set of data is transferred to the buffer. Therefore, it would take less than 2 min to determine the activation times at all the sites on the ventricles. With the manual method, it requires more than 45 min to obtain the activation times at these sites.

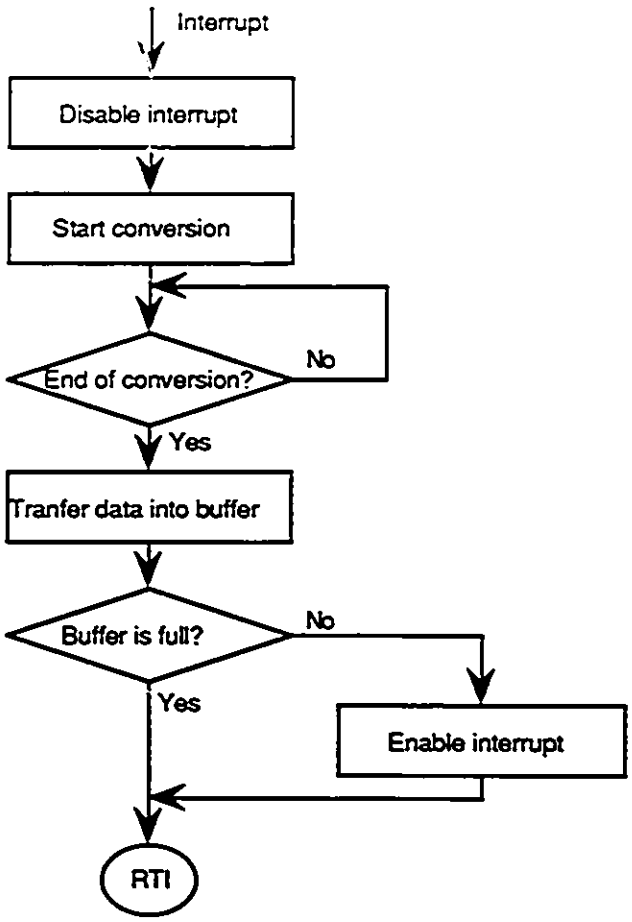

Figure 4.5 Flowchart of data-acquisition control process.

*** Mapping module:**

This module provides the interface between the user and the cardiac mapping system to performs the following functions:

- To receive the request from the user.
- To display and to store the numerical values of the activation times and the waveforms of the signals at a specific location on the ventricles. A vertical time marker is also displayed on each waveform to show where the activation time is detected.
- To map the activation time at different locations on the ventricles.

- To permit manual correction of the computer-estimated activation times.

The flow chart of this module is shown in Fig. 4.6. and its program listing is shown in Appendix B.

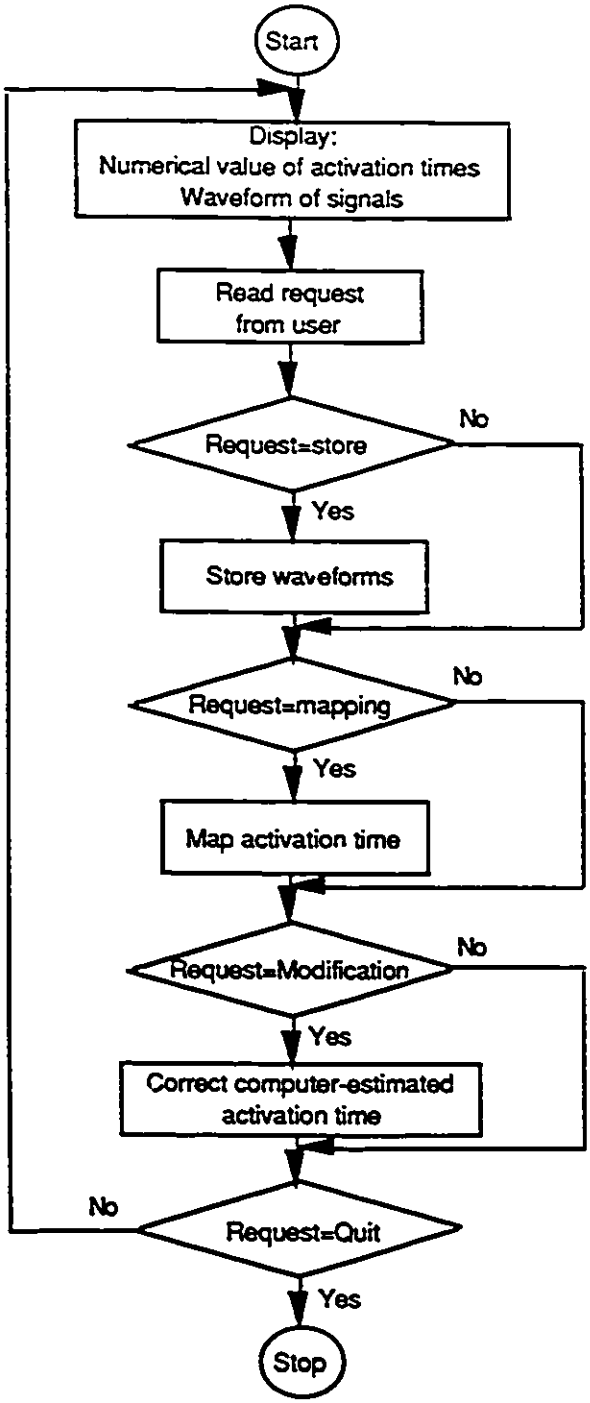

Figure 4.6 Flowchart of mapping module.

* Data-retrieval module:

This module provides the user the facility to retrieve the results and the waveform for further study after the operation is complete. Since the waveforms at a specific location on the ventricles were stored as a set of 1200 samples, the user can request a display of the waveform at a specific set and sample. Please see Appendix C for program listing. The flow chart of this module is shown in Fig. 4.7.

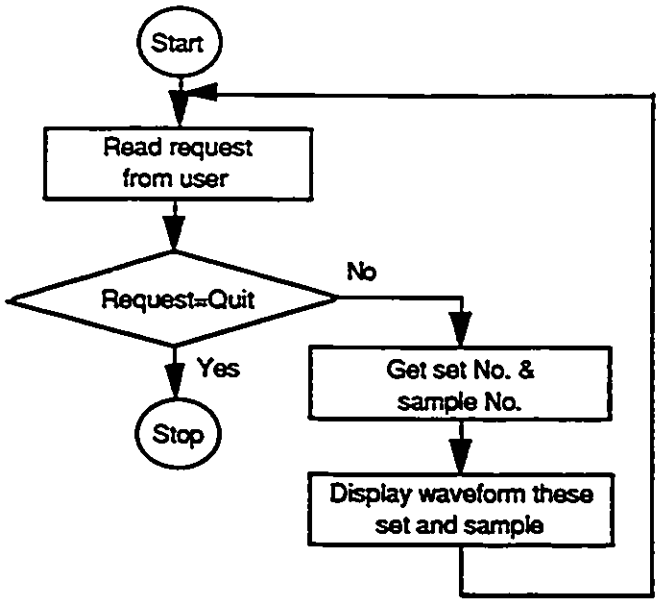

Figure 4.7 Flowchart of data-retrieval module.

CHAPTER 5

NUMERICAL AND EXPERIMENTAL RESULTS

5.1 Simulation results:

Before the computerised cardiac mapping system was used on the patients' data, it was first tested on a simulator. The simulator is a simple board that can generate different analog signals which are similar to those measured from a patient. These signals are then fed into the computerized cardiac mapping system to see if it can detect the activation time properly. The simulator was used to compare the results of different methods of analysis.

5.1.1 The hardware configuration of the simulator:

The diagram of the simulator is shown in Fig. 5.1.

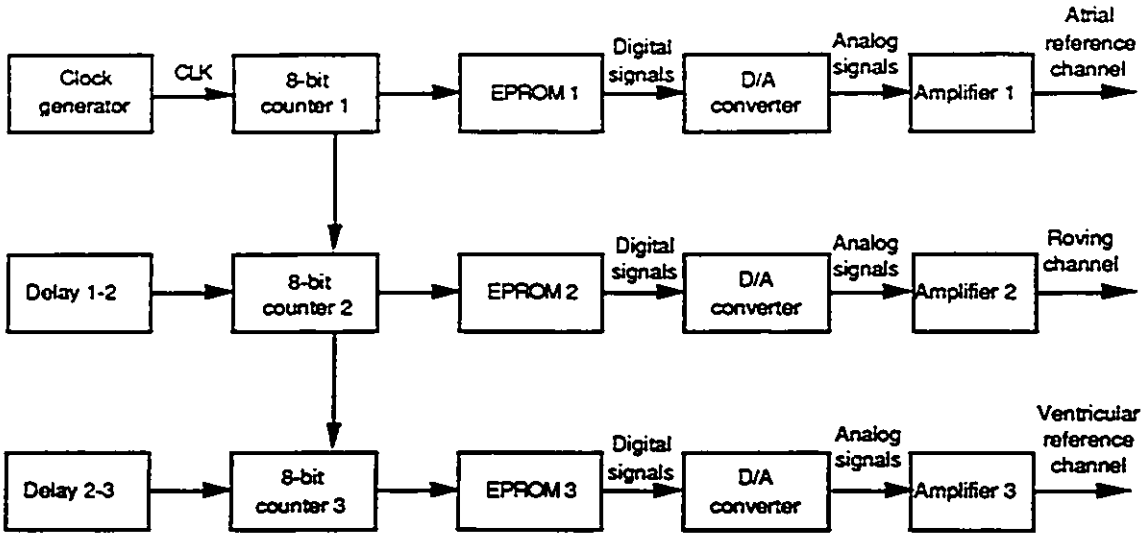

Figure 5.1 Diagram of the simulator.

A clock generator produces one clock pulse every 2 ms. A desired waveform is digitized and stored in the EPROM which has 256 samples. Therefore, the output waveform has period of 512 ms or a frequency of 2 Hz which is equal to the pacing frequency used in the operating room.

The activation potentials at the roving channel occur approximately from 100 ms to 300 ms after the activation of the atrial reference channel (or the ventricular reference channel). Delay 1-2 is used to set the local activation time of the roving channel relative to the atrial reference channel and Delay 2-3 is used to set the local activation time of the roving channel relative to the ventricular reference channel.

The output digital signals from the EPROMs are then converted into analog signals by the D/A converters. The analog signals are then amplified to specific values which are at the same voltage levels as those obtained from the patients.

5.1.2 Simulation results:

The system has successfully detected many different types of waveforms as shown in Figures 5.2 - 5.11 (The arrow in each figure indicates where the activation occurs). The proposed algorithm correctly detected the time of local activation even with the signals that were severely corrupted by noise. These waveforms were digitized and stored in the EPROMs of the simulator. The output analog signals were passed into the system for detecting the activation time. The three other algorithms discussed in chapter 1 were also implemented to detected the activation time of these waveforms. The results presented in Table 5.1 demonstrates that the proposed algorithm has greater number of true detections than these three methods.

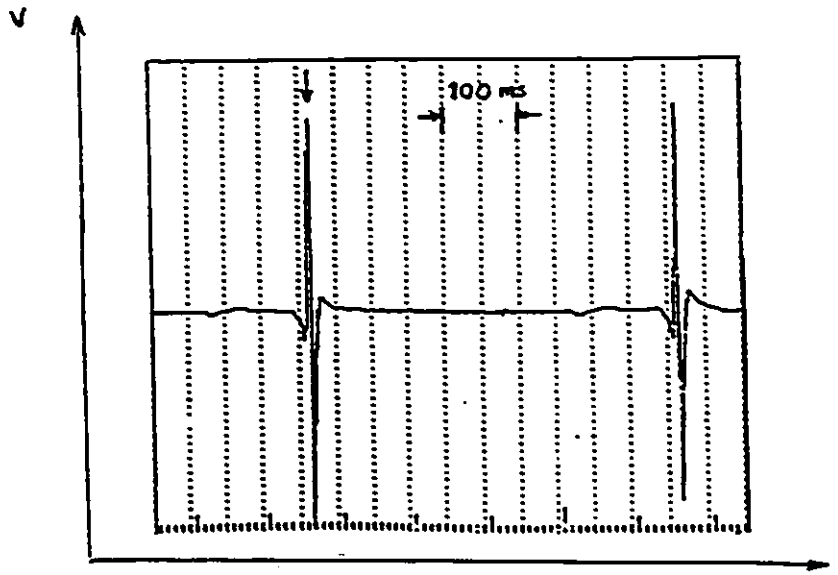

Figure 5.2 Waveform 1.

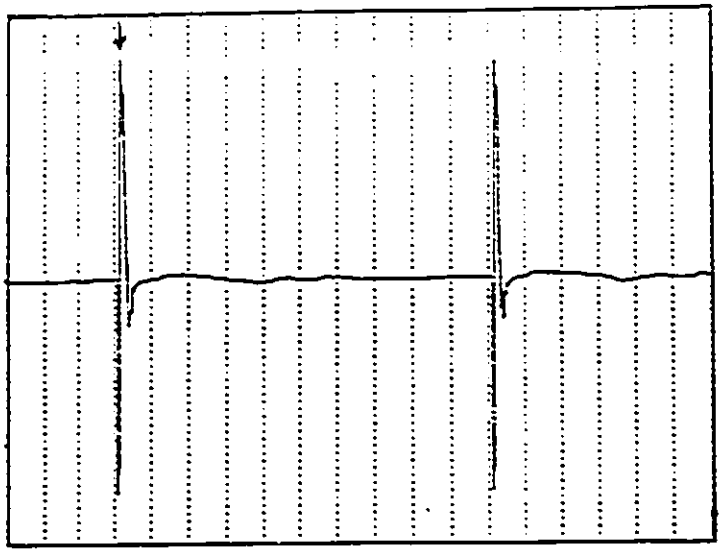

Figure 5.3 Waveform 2.

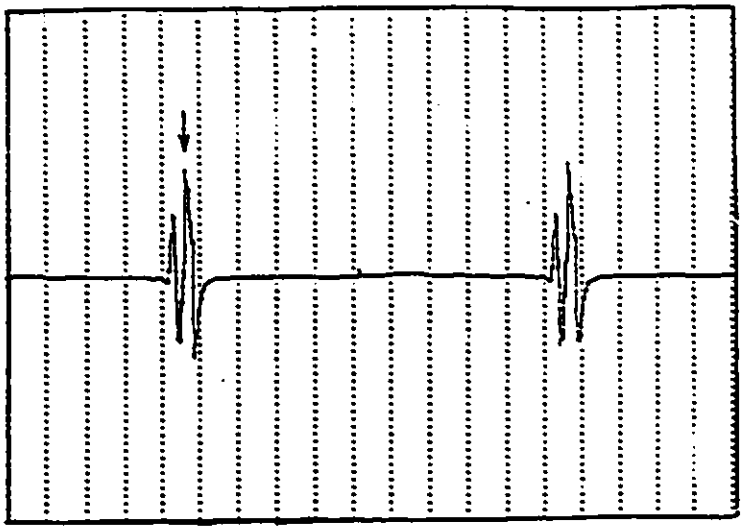

Figure 5.4 Waveform 3.

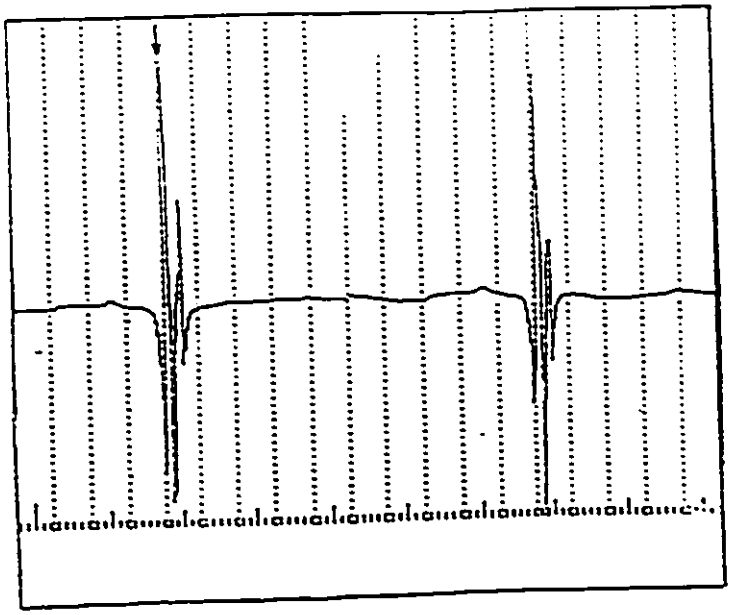

Figure 5.5 Waveform 4.

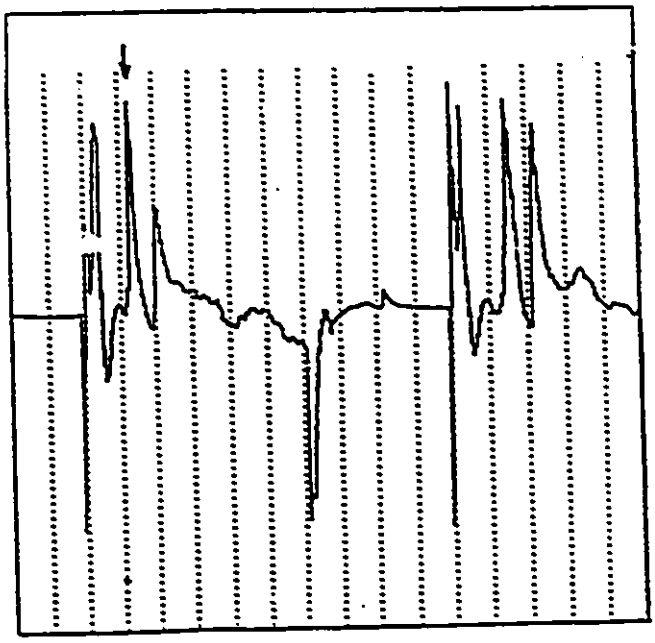

Figure 5.6 Waveform 5.

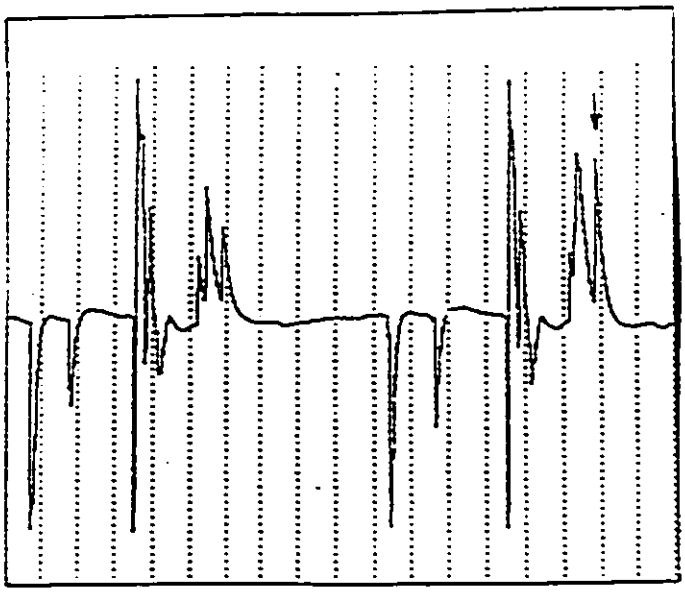

Figure 5.7 Waveform 6.

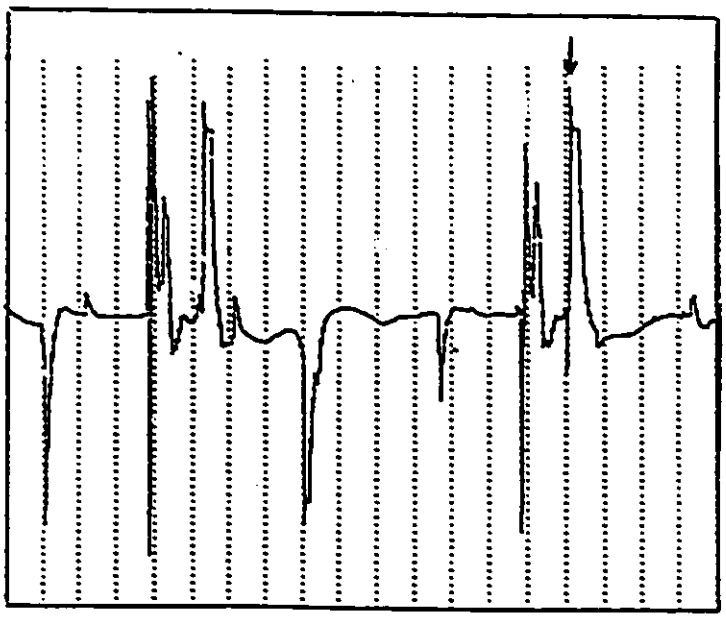

Figure 5.8 Waveform 7.

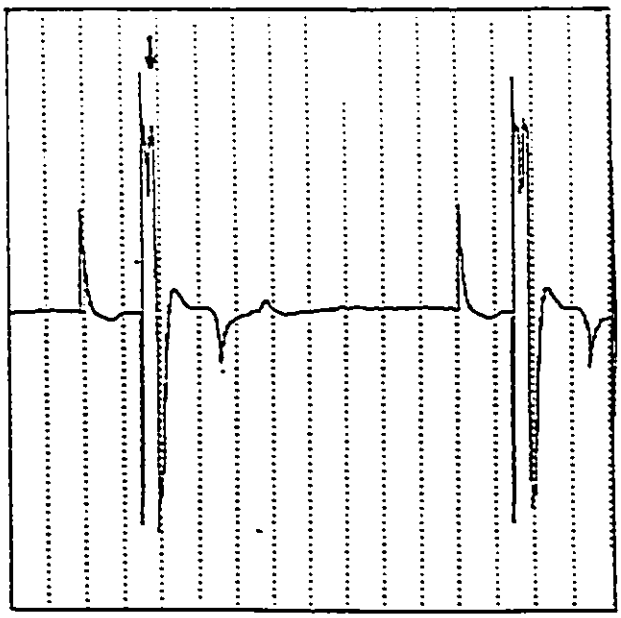

Figure 5.9 Waveform 8.

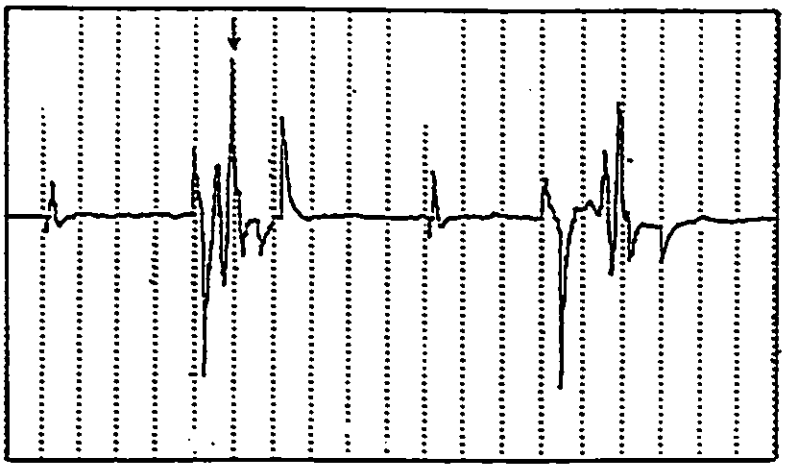

Figure 5.10 Waveform 9.

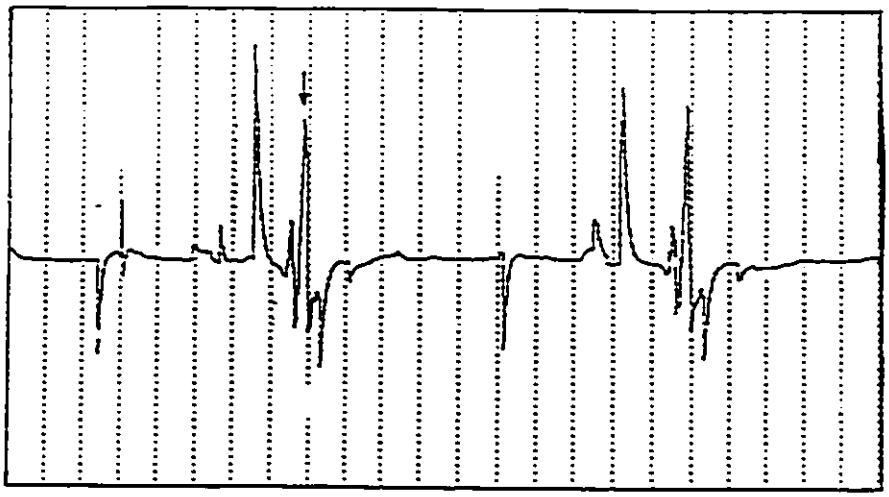

Figure 5.11 Waveform 10.

In the following experiment, true detection was considered to have occurred when the estimated activation time was within +/- 5 ms of the true activation time, and false detection was any detection that was greater than +/- 5 ms away from the true activation time. From the results in Table 5.1, we see that with the clean signals as in Waveforms 1-4, all the methods gave true detection. When the signals contained many sources of noise, the simple method failed to give true detection. The data-fitting method gave large number of true detections, but it required much more computation time and difficult to implement for large number of electrodes. The proposed algorithm had a greater number of true detections (100%) than the other three methods.

Methods / Waveforms	Simple method	Interpolation n=3	Data-fitting n=5	Proposed method
Waveform 1	TD	TD	TD	TD
Waveform 2	TD	TD	TD	TD
Waveform 3	TD	TD	TD	TD
Waveform 4	TD	TD	TD	TD
Waveform 5	FD	TD	TD	TD
Waveform 6	FD	TD	TD	TD
Waveform 7	FD	FD	TD	TD
Waveform 8	FD	FD	FD	TD
Waveform 9	FD	FD	FD	TD
Waveform 10	FD	FD	FD	TD
Sensitivity	4/10	6/10	7/10	10/10

TD: True detection

FD: False detection

Table 5.1 The number of true detections of each method using 10 different waveforms.

5.2 Clinical results:

The computerized cardiac mapping system was also used directly in the operating room to map the activation times of the patients. Tables 5.2 -5.4 show the results obtained from three different patients.

The values of the activation times obtained from the current mapping system at the University of Ottawa Heart Institute are also listed. In this system, the activation times were measured manually on the signals recorded on the chart paper, the resolution is +/-10 ms. One can see the differences in the results obtained using these two systems.

Patient 1 / Site	Atrial pacing frequency: 2 Hz (or period: 500 ms)		Ventricular pacing frequency: 2 Hz (or period: 500 ms)	
	Manual results (ms)	Computer results (ms)	Manual results (ms)	Computer results (ms)
A	230	233	220	210
B	230	227	240	237
C	240	240	270	273
D	260	262	280	275
E	240	243	270	264
F	250	249	250	252
G	230	233	240	246
H	190	192	205	202
J	160	164	190	198
K	160	161	190	181
L	150	150	190	190
M	120 -> 150	118 ->150	170 -> 190	168 -> 190
N	140	139	190	188
O	160	150	210	210
P	180	175	250	254
Q	220	215	260	262

Table 5.2 The values of the activation times obtained from the first patient using two different systems.

The manual results and the computer results both showed that site M on the left ventricle of patient 1 was early. After ablation, both results also showed that the activation time of site M was corrected.

Patient 2 / Site	Atrial pacing frequency: 2 Hz (or period: 500 ms)		Ventricular pacing frequency: 2 Hz (or period: 500 ms)	
	Manual results (ms)	Computer results (ms)	Manual results (ms)	Computer results (ms)
A	230	234	220	225
B	215	223	220	219
C	230	224	230	230
D	190	178	190	188
E	200	217	210	204
F	240	248	220	217
G	185	195	190	190
H	190	198	190	189
J	180	171	165	162
K	140 -> 170	151 -> 171	110 -> 160	112 -> 151
L	170	169	180	183
M	180	175	180	175
N	190	191	180	176
O	190	198	190	179
P	190	186	200	186
Q	210	217	200	197

Table 5.3 The values of the activation times obtained from the second patient using two different systems.

The manual results and the computer results both showed that site K on the left ventricle of patient 2 is early. After ablation, both results also showed that the activation time of site K was corrected.

Patient 3 / Site	Atrial pacing frequency: 2 Hz (or period: 500 ms)		Ventricular pacing frequency: 2 Hz (or period: 500 ms)	
	Manual results (ms)	Computer results (ms)	Manual results (ms)	Computer results (ms)
A	200	205	200	200
B	220	223	200	210
C	240	250	230	232
D	270	277	240	240
E	174	180	200	204
F	170	166	180	190
G	180	178	190	190
H	180	178	200	206
J	160	158	200	188
K	160	151	170	174
L	120 -> 160	125 -> 158	150 -> 160	154 -> 164
M	180	185	180	190
N	160	156	170	170
O	160	162	180	183
P	210	217	200	200
Q	200	190	200	210

Table 5.4 The values of the activation times obtained from the third patient using two different systems.

The manual results and the computer results both showed that site L on the left ventricle of patient 3 is early. After ablation, both results also showed that the activation time of site L was corrected.

In order to assess the validity of the proposed method, the following tests were used to evaluate the reliability of the system:

- Determination of earliest activation site.
- Paired t test [32].
- Spearman rank correlation test [33].

Determination of earliest activation site:

To determine the site of ablation, the surgeon needs to know which sites have early activation. From the results shown in the above tables, it is evident that the computerized mapping system detected the same site of earliest activation as well as the manual method for all three patients. The differences in activation time between the ablation site and the next minimum activation time were 29±14 ms (mean±σ). This confirms that a precision of measurement of +/- 5 ms is adequate for the task of localization of the ablation site.

The data were further examined by using a paired t test and Spearman rank correlation test. Tables 5.5 -5.10 give the analysis used in these tests:

Patient 1 Atrial Site	Manual results (ms) x	rank r1	Computer results (ms) y	rank r2	x-y (ms)	(r1-r2)^2
A	230	11	233	11.5	-3	0.25
B	230	11	227	10	3	1
C	240	13.5	240	13	0	0.25
D	260	16	262	16	-2	0
E	240	13.5	243	14	-3	0.25
F	250	15	249	15	1	0
G	230	11	233	11.5	-3	0.25
H	190	8	192	8	-2	0
J	160	5	164	6	-4	1
K	160	5	161	5	-1	0
L	150	3	150	3.5	0	0.25
M	120	1	118	1	2	0
N	140	2	139	2	1	0
C	160	5	150	3.5	10	2.25
P	180	7	175	7	5	0
C	220	9	215	9	5	0

t test:

Mean difference = 0.56 ms

Standard error = 0.94 ms

t = 0.60

Spearman correlation:

rs = 0.99

Table 5.5 The results of t-test and Spearman correlation using the atrial data from patient 1.

Patient 1 Vent / Site	Manual results (ms) x	rank r1	Computer results (ms) y	rank r2	x-y (ms)	(r1-r2)^2
A	220	8	210	7.5	10	0.25
B	240	9.5	237	9	3	0.25
C	270	14.5	273	15	-3	0.25
D	280	16	275	16	5	0
E	270	14.5	264	14	6	0.25
F	250	11.5	252	11	-2	0.25
G	240	9.5	246	9	-6	0.25
H	205	6	202	6	3	0
J	190	3.5	198	5	-8	2.25
K	190	3.5	181	2	9	2.25
L	190	3.5	190	4	0	0.25
M	170	1	168	1	2	0
N	190	3.5	188	3	2	0.25
C	210	7	210	7.5	0	0.25
P	250	11.5	254	12	-4	0.25
C	260	13	262	13	-2	0

t test:

Mean difference = 0.94 ms

Standard error = 1.27 ms

t = 0.74

Spearman correlation:

rs = 0.99

Table 5.6 The results of t-test and Spearman correlation using the ventricular data from patient 1.

Patient 2 Atrial Site	Manual results (ms) x	rank r1	Computer results (ms) y	rank r2	x-y (ms)	(r1-r2)^2
A	230	14.5	234	15	-4	0.25
B	215	13	223	13	-8	0
C	230	14.5	224	14	6	0.25
D	190	8	178	5	12	9
E	200	11	217	11.5	-17	0.25
F	240	16	248	16	-8	0
G	185	5	195	8	-10	9
H	190	8	198	9.5	-8	0.25
J	180	3.5	171	3	9	0.25
K	140	1	151	1	-11	0
L	170	2	169	2	1	0
M	180	3.5	175	4	5	0.25
N	190	8	191	7	-1	1
C	190	8	198	9.5	-8	2.25
P	190	8	186	6	4	4
C	210	12	217	11.5	-7	0.25

t test:

Mean difference = -2.81 ms

Standard error = 2.05 ms

t = -1.37

Spearman correlation:

rs = 0.96

Table 5.7 The results of t-test and Spearman correlation using the atrial data from patient 2.

Patient 2 Vent Site	Manual results (ms) x	rank r1	Computer results (ms) y	rank r2	x-y (ms)	(r1-r2)^2
A	220	14	225	15	-5	1
B	220	14	219	14	1	0
C	230	16	230	16	0	0
D	190	6.5	188	8	2	2.25
E	210	12	204	12	6	0
F	220	14	217	13	3	1
G	190	7.5	190	10	0	6.25
H	190	7.5	189	9	1	2.25
J	165	2	162	2	3	0
K	110	1	112	1	-2	0
L	180	4	183	6	-3	4
M	180	4	175	3	5	1
N	180	4	176	4	4	0
C	190	7.5	179	5	11	6.25
P	200	10.5	186	7	14	12.25
C	200	10.5	197	11	3	0.25

t test:

Mean difference = 2.69 ms

Standard error = 1.21 ms

t = 2.23

Spearman correlation:

rs = 0.95

Table 5.8 The results of t-test and Spearman correlation using the ventricular data from patient 2.

Patient 3 Atrial Site	Manual results (ms) x	rank r1	Computer results (ms) y	rank r2	x-y (ms)	(r1-r2)^2
A	200	11.5	205	12	-5	0.25
B	220	14	223	14	-3	0
C	240	15	250	15	-10	0
D	270	16	277	16	-7	0
E	174	7	180	9	-6	4
F	170	6	166	6	4	0
G	180	9	178	7.5	2	2.25
H	180	9	178	7.5	2	2.25
J	160	3.5	158	4	2	0.25
K	160	3.5	151	2	9	2.25
L	120	1	125	1	-5	0
M	180	9	185	11	-5	4
N	160	3.5	156	3	4	0.25
O	160	3.5	162	5	-2	2.25
P	210	13	217	13	-7	0
Q	200	11.5	190	15	10	2.25

t test:

Mean difference = -1.06 ms

Standard error = 1.48 ms

t = -0.72

Spearman correlation:

rs = 0.97

Table 5.9 The results of t-test and Spearman correlation using the atrial data from patient 3.

Patient 3 Vent / Site	Manual results (ms) x	rank r1	Computer results (ms) y	rank r2	x-y (ms)	(r1-r2)^2
A	200	10	200	9.5	0	0.25
B	200	10	210	13.5	-10	12.25
C	230	15	232	15	-2	0
D	240	16	240	16	0	0
E	200	10	204	11	-4	1
F	180	5	190	6.5	-10	2.25
G	190	7	190	6.5	0	0.25
H	200	10	206	12	-6	4
J	200	10	188	5	12	25
K	170	2.5	174	3	-4	0.25
L	150	1	154	1	-4	0
M	180	5	190	8	-10	9
N	170	2.5	170	2	0	0.25
C	180	5	183	4	-3	1
P	200	10	200	9.5	0	0.25
C	200	10	210	13.5	-10	12.25

t test:

Mean difference = -3.19 ms

Standard error = 1.41 ms

t = -2.26

Spearman correlation:

rs = 0.90

Table 5.10 The results of t-test and Spearman correlation using the ventricular data from patient 3.

t-Distribution test:

Although the computerized mapping system detected the same earliest activation site as the manual method, there are slight differences in the activation times between two methods. With the assumption that the distribution of the differences is normal and the population mean μ_0 and the standard deviation σ are unknown, the paired t test is used to test the hypothesis H_0 ($\mu_0 = 0$).

For all six cases:

1. Statistical test: The Student t-ratio is chosen on the assumption that the distribution of differences is a normally distributed variable in which σ is unknown.
2. Null hypothesis (H_0): The population mean $\mu_0 = 0$ ms.
3. Alternate hypothesis (H_1): The population mean $\mu_0 \neq 0$ ms.
4. Significant level: $\alpha = 0.05$
5. Sampling distribution: The sampling distribution is the Student t-distribution with df=15.
6. Critical region: Since H_1 is nondirectional, the critical region consists of all values of t >= +2.131 or t<= -2.131 (from Table E in [32]).

The calculated t-ratio for all six cases are shown in Tables 5.5-5.10.

Although the null hypothesis is rejected in two cases, the mean difference in these two cases is approximately 3 ms which is still smaller than the resolution of the manual method. A better approximation of μ_0 may be obtained with a larger sample size [33]. If we assume that the distribution of the difference is the same for all six cases, we can perform the t-ratio test on a group of 96 samples (df=95). The result of this test:

- Sample mean difference $\bar{x} = -0.48$ ms.
- Standard error $S_{\bar{x}} = 0.61$ ms.
- Sample standard deviation S = 5.95 ms
- t = -0.79.
- Critical region: t >= +2.00 or t <= -2.00 (from Table E in [32]).

Using the paired t-test, the 95 % confidence interval for μ_0 is (-1.63 ms, 0.81 ms) [34]. The 95% confidence interval for standard deviation σ_0 is (5.21 ms, 6.93 ms) [35]. Thus, the population mean μ_0 is not significantly different from zero and the standard deviation σ_0 is smaller than the resolution of the manual method (10 ms).

Spearman correlation:

A further test which may be applied to the data is to examine the correlation between the results from each method. Because these data are not normally distributed, the Spearman non parametric rank correlation test was used.

For all six cases:

1. H_o: The correlation between two method is zero.
2. H_1 : The correlation between two method is not zero.
3. Test statistic: r_s.
4. Significant level: $\alpha = 0.05$.
5. Critical region: $|r_s| > 0.506$ (from Table G in [32]).

The Spearman correlation factors shown for all cases in Tables 5.5 -5.10 are greater than critical values. Thus, H_o is rejected in all cases. From these results, we see a very high positive correlation between the two methods. Therefore, we can conclude that the computerized mapping system provides comparable results with the manual method.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

6.1 Discussion:

Although this thesis has examined a method for analysis of single channel data, the results are applicable to any mapping system. The computerized cardiac mapping system was found more powerful and versatile than the existing system. The total time required to obtain all data and to construct the map is less than 2 min because the activation time at each location can be obtained within 1.5 s. The display mode which uses different colors for different ranges of activation enables rapid recognition of areas with early activity or delayed potentials. The color at each location on the ventricles represents the activation time at this location. Therefore, the surgeons can quickly localize the areas causing the arrhythmia during the operation. Although we cannot save the map, the signals recorded at each location and its numerical activation time are stored in the disk for post-operation study. The map of the activation times is shown in Appendix E.

In many computerized cardiac mapping system that have been developed [13, 14], the data-analysis process is done after the data-acquisition. Thus, it takes longer to obtain the results. In the proposed system, the time during which the data-acquisition and data-analysis are processed concurrently is approximately 1.2 s for each site on the ventricles. The time taken for both processes to obtained the final result is about 1.5 s. Therefore, the efficiency of this system is considerably high.

Although the activation time detection algorithm appears complicated, it is reliable as shown in Chapter 5: the number of true detections over the total number of detections (the detector sensitivity) is greater than those of other methods [7, 20, 22].

The resolution of this system is better than that of the manual system because the smallest scale on the chart paper is 10 ms. The largest mean difference is 3 ms which is smaller than the error of the manual method. The 95 % confidence interval for μ_0 is (-1.63 ms, 0.81 ms) which is not significantly different from zero and the 95 % confidence interval for σ_0 is (5.21 ms, 6.93 ms) which is still smaller than the error of the manual method. The Null hypothesis H_0 in Spearman rank correlation test is rejected. Therefore, we can see a very high positive correlation between these two methods.

Finally, the cost of the system is relatively low as it requires only a personal computer instead of large computers as in other systems [8, 13, 16, 18].

*** Proposals:**

Studies of the computerized cardiac mapping system will continue while the fundamental information is being developed. This system can be further improved as more knowledge and experience are gained, just as other systems have been modified many times in the past. The interesting extension of the studies of the computerized cardiac mapping system would be the one or more of the followings:

(1) New pacing method:

As mention in chapter 2, the local activation at the roving electrode usually happens in about 350 ms after the activation at the reference channel takes place. In order to detect the local activation, first we must detect the activation at the reference channel.

The pacing signal frequency at the reference channel that can be applied to a patient, depends on the ability of a patient's cells to conduct the electrical activity. Some can conduct very fast up to one stimulus every 200 ms. Some are much slower than this. However, the slowest pacing frequency that a patient would have is about the normal pacing frequency of the SA node (the natural pace maker of the heart). Hence, the slowest pacing frequency can be 70 beats per minute or one stimulus in every 850 ms.

Right now, the pacing device and the mapping system are completely separated from each other. In order to have one complete beat at the reference channel we need continuous 850 ms of data plus another 350 ms to detect the activation at the roving electrode. That is, we need to store at least 1200 ms of data to process. (See Fig. 6.1a). Therefore, we can obtain the result of the map every second heart beat instead of every heart beat.

It would be better to connect the pacing device to the mapping system so that, whenever a stimulus is generated, the mapping system would start to store only 350 ms of data for processing (See Fig. 6.1b). Thus, we do not have to store 850 ms at the reference channel in order to detect the activation at the reference channel. This can reduce the processing time down to three quarters and the results can be obtain at every heart beat.

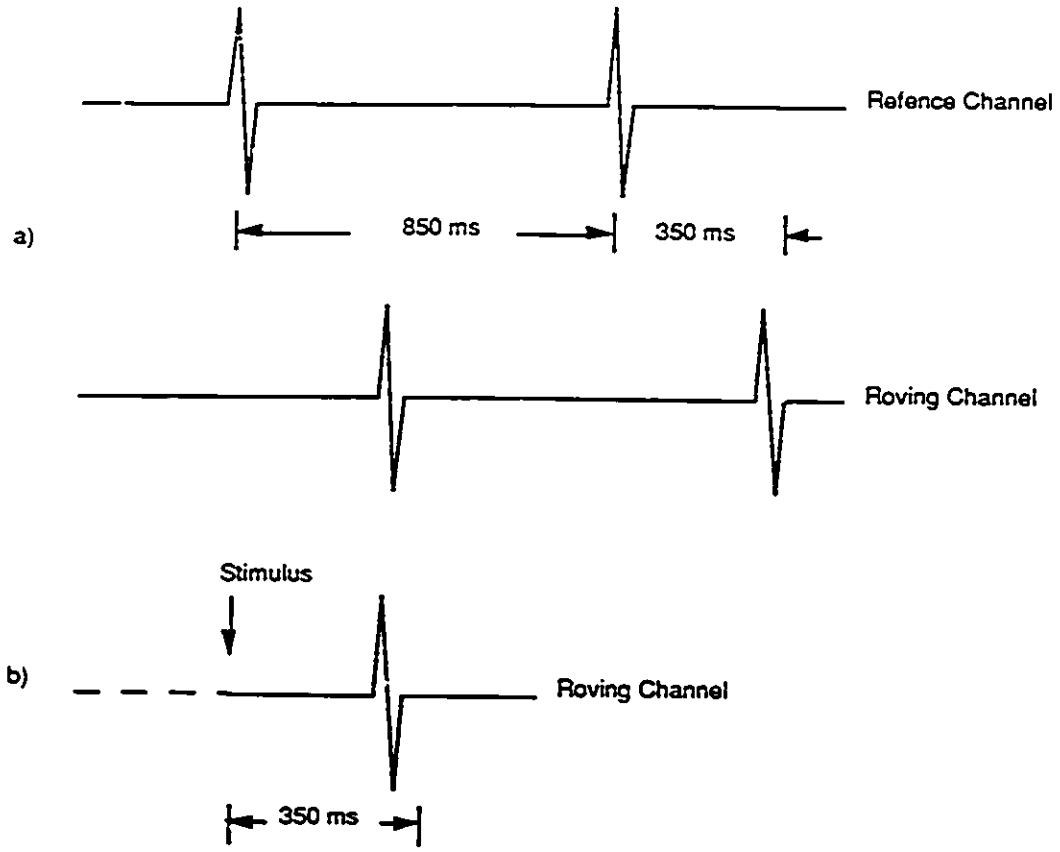

Figure 6.1 a) The mapping system and the pacing device are not connected.
b) The mapping system and the pacing device are connected.

(2) Multiple-electrodes:

This system is implemented with only three electrodes, but it can be expanded to many more electrodes so that the data at many location on the ventricles can be recorded at the same time. For example, a band of many electrodes can be quickly and easily applied to the heart and allows electrograms from many sites on ventricles to be recorded simultaneously during one cardiac cycle.

If each epicardial point is recorded during a different cardiac cycle, it is difficult of find the cause of rapid, unstable arrhythmias [11, 23]. With the band of many electrodes, we can overcome these limitations by capturing all the electrophysiologic data necessary for constructing a map of left or right ventricle within one cardiac cycle. At the present time, recordings are made at only 17 regions labelled from A to Q on the ventricles with the assumption that the potentials of five areas labelled from 1 to 5 in each region are the same. When the number of electrodes increases, one can measure the electrical activity at many smaller areas within each region as shown in Fig. 2.3. This would help the surgeons to localize more precisely the area causing the arrhythmia.

Even if the electrodes are fixed to a band, it is easy to place the electrodes over the standard predetermined points of the ventricles without losing the precision of an electrode band. However, the size of the heart varies from patient to patient. A standard band may introduce some errors in location. Although these errors in location may distort the map of activation times [23], they do not affect identification of the site of earliest recorded activation directly on the epicardium.

(3) Variable gain:

In the variable gain recording system, the signals have approximately similar amplitudes after optimum amplification. Thus, we do not know the real potentials of each signal. If the gain is recorded simultaneously with the recorded signal, it would be easy to differentiate which signals have large or small amplitude. In order to detect the activation in each channel, we do not have to know the potential of its signals. However, the potentials of the signals may be used to classify the myocardium as normal or infarcted [27].

6.2 Conclusions:

The technique used to record the signals from the ventricles is either single roving electrode or multiple electrodes. Several activation detection algorithms have been suggested but they may not prove to be the algorithm of choice in all cases due to their unreliability or the increased computation time required.

The proposed cardiac mapping system was developed to analyse single channel data but it can be expanded to handle multiple channels. Its efficiency is high because the data-acquisition and data-analysis are done concurrently. Furthermore, the proposed activation detection algorithm does not require large computation time. Therefore, it is useful to meet the time constraints in the operating theatre. The reliability of this algorithm with the rapid display of results helps the surgeons to localize the area of myocardium responsible for the arrhythmia. Finally, the cost of the system is low since it was developed using a personal computer instead of a more powerful computer. Overall, the system has been developed and proven to be successful in clinical environment for its contributions to the ventricular arrhythmia treatment since it is a useful tool for the intraoperative study of patients who undergo surgery for drug resistant, life threatening ventricular arrhythmias.

An interesting expansion of the system could be using the pacing stimulus to initiate the data-acquisition process or using multiple electrodes to record simultaneously data from many sites on the ventricles.

APPENDIX A

ACTIVATION TIME DETECTION PROGRAM LISTING

```
/*****************************************************************************
**                                                                          **
**  Name of module: Actime.h                                                **
**                                                                          **
**  This module contains the header and the global declarations             **
**  for the module Actime.c                                                 **
**                                                                          **
**  Written by: Thuthuy Bach                                                **
**                                                                          **
**  Starting date: May 10 1990                                              **
**                                                                          **
**  Closing date:  Nov 15 1990                                              **
**                                                                          **
*****************************************************************************/

/* The following libraries must be included */

#include <dos.h>
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#include <math.h>
#include <process.h>

#define REFCH  0            /* Reference channel */
#define CH1    1            /* Roving channel */
#define CH2    2            /* Co-reference channel */
#define MAXCH  3            /* Maximum number of channel */
#define MAXSP  1200         /* Maximum number of samples */

#define AD      0x240       /* address of A/D board */
#define MSBAD   AD+0x08     /* MSB of result from A/D board */
#define LSBAD   AD+0x09     /* LSB of result from A/D board */
#define TIME0   AD+0x10     /* address of counter 0 */
#define TIMECT  AD+0x13     /* address of timer controller */
#define PORTA   AD+0x0c     /* address of output port A */
#define PORTB   AD+0x0d     /* address of output port B */
#define PORTC   AD+0x0e     /* address of output port C */
#define PORTCT  AD+0x0f     /* address of output port controller */
#define VECINT  8           /* number of vector interrupt */
#define INT     3           /* choose interrup 3 */
#define MASAP   0x21        /* mask interrupt */
#define CT8259  0x20        /* control port   */

#define fn "c:test.dat"   /* data file that stores all the data */
                          /* from the patient                  */
typedef struct {          /* data at each location on the ventricles */
char *let;                /* location */
int  xp;
int  yp;
int  t1;                  /* difference in activation time between  */
                          /* atrial reference and roving channels */
int  t2;                  /* difference in activation time between */
                          /* ventricular reference and roving channels */
}bb;

void main(void);          /* main program */
void foreword(void);      /* introduction of the mapping system */
```

```
void InitMap(void);          /* Initialize the mapping system */
void DisplayMap(void);       /* display the ventricles */
void menu(void);             /* menu of the map */
void InitFile(void);         /* initialize patient's data file */
void upsig(void);            /* display the signal */
void keyin(void);            /* read the key board */
void ucmap(void);            /* uccate the map */
void CoRuler(void);          /* display color with corresponcing */
                             /* activation time */
void scr(void);              /* modification of the activation time */
void cursor(int,int);        /* move the cursor in the same wincow */
int  wcow(int,int);          /* move the cursor to next window */
void move(int);              /* move the cursor up or down */
void jump(int);              /* place the cursor at the right location */

void  actime(void);          /* find the activation times at all channels */
void  InitAD(void);          /* initialize the A/D board */
int   ADConvert(int);        /* digitize the signals */
void  StoreData(void);       /* store all the samples */
void  PeakRef(void);         /* find the maximum amplitude at the */
                             /* reference channel */
int   FindSlop(int,int,int);     /* find maximum positive and negative slope */
int   FindPeak(int,int,int,int); /* find maximum amplitude at other channels */
void  EnbInt(void);              /* enable the interupt */
void  DisInt(void);              /* disable the interupt */
void  SeTimer(unsigned int, unsigned int);  /* set the timer */
void  CheckTimer(void);          /* check for the timer */
void  search(void);              /* detect the activation */
void  slpwcow(int,int,int);      /* find maximum positive and negative */
                                 /* slopes in the window */
void  message(char*);            /* send a message */
void  clrmsg(int,int,int,int);   /* clear a message */
void  interrupt sampling(void);  /* interupt at every sample */

int data[MAXCH][MAXSP];          /* data array at all channels */
int POSLP,       /* maximum positive slope */
    WHPOS,       /* sample where maximum positive slope occurs */
    NESLP,       /* maximum negative slope */
    WHNEG,       /* sample where maximum negative slope ossurs */
    csc,         /* current sample */
    dreq,        /* data aquisition required */
    clet,        /* current location on the ventricles */
    peak[MAXCH], /* maximum peak at a channel */
    MAXI[MAXCH], /* maximum amplitude at a channel */
    MINI[MAXCH]; /* minimum amplitude at a channel */
FILE *dat;       /* data file of patient */
```

```
/*****************************************************************
**                                                              **
**  Name of module: actime.c                                    **
**                                                              **
**  This module detects the activation at each channel          **
**                                                              **
**  Written by: Thuthu, Bach                                    **
**                                                              **
**  Starting date: May, 10 1990                                 **
**                                                              **
**  Closing date:                                               **
**                                                              **
*****************************************************************/
#include "cimcat.h"     /* reading module */

/*******************/

void actime()

/*******************/

/* This routine digitizes the signals, stores the data and detects */
/* the activation at each channel                                  */
{
    StoreData();  /* digitize and store the signal in a buffer */
    PeakRef();    /* detect the activation at the reference channel */
    search();     /* detect the activation at other channels */
}


/*******************/

void InitAD()

/*******************/

/* This routine initializes the A/D board */
{
  csp=0;                  /* reset the current sample */
  outportb(TIMECT,0x32 );  /* load timer controller: counter 0 mode 0 */
  outportb(PORTCT,0x82 );  /* port controller: A,B and C are input ports */
}

/*******************/

int  ADConvert(chan)
int chan;

/*******************/

/* This routine reads the digitized signals from the A/D board */
{
  int loop;
  unsigned int EOC;

  outportb(AD+chan),0 );        /* assign a specific channel at A/D board */
  for (loop=0;loop < 5; ++loop );  /* wait until this channel is assigned */
```

```
  outportb(MSB4D,0);          /* reset the output of the A/D port */
  EOC = 0;                    /* reset the EOC bit */

  while (EOC==0)              /* check the EOC bit */
    EOC=inportb(PORTB)&128;

 /* read a digital signal */
  return( (inportb(MSB4D)<<4)+(inportb(LSB4D)>>4)-2048);
}


/******************/

void StoreData()

/******************/

/* The functions of this routine:                                    */
/*     - To start sampling all the channels.                         */
/*     - To find the maximum positive and negative slope at reference */
/*       channel in the first 650 samples.                           */
/*     - To find the peak of the activation for the reference channel */
/*       in the first 650 samples.                                   */
/*     - To find the maximum and minimum values at all channels.     */

{
  int samp1,chan1,botpos,topneg,up,temp,slp;

  /* initialize the following variables to 0 */

  MINI[0]=MINI[1]=MINI[2]=2047;
  MAXI[0]=MAXI[1]=MAXI[2]=-2048;
  topneg=botpos=0;
  POSLP=WHPOS=0;
  NEGLP=WHNEG=0;
  up=0;

  StrInt();     /* start the sampling */

  /* Data analysis after each sample is collected */

  while (csp<=MAXSP)
  {
    temp=csp;

    /* find the maximum positive and negative slopes at reference channel */
    /* in the first 650 samples.                                          */

    if ((temp>1)&&(temp<651))
    {
      /* down slope */
      if (data[0][temp-1]<=data[0][temp-2])
      {
        slp=0;

        /* if down slope right after up slope find positive slope */
        if ((up==1)&&(temp-2-botpos!=0))
        {   slp=(data[0][temp-1]-data[0][botpos])/(temp-2-botpos);
```

```
            if (slp>POSLP) { POSLP=slp; w=POS=temp-2; }
          }

          totpos=temp-1;
          up=0;
        }

        /* up slope */
        if (data[0][temp-1]>data[0][temp-2])
        {
          slp=0;

          /* if up slope right after down slope find negative slope */
          if ((up==0)&&(temp-2-topneg!=0))
          { slp=(data[0][temp-2]-data[0][topneg])/(temp-2-topneg);
            if (slp<NEGLP) { NEGLP=slp; WHNEG=topneg; }
          }
          topneg=temp-1;
          up=1;
        }
      }

      /* find maximum and minimum values at all channels */
      if (temp>0)
      for (chan1=0;chan1<MAXCH;chan1++)
      { if (data[chan1][temp-1]<MINI[chan1])
          MINI[chan1]=data[chan1][temp-1];
        if (data[chan1][temp-1]>MAXI[chan1])
        {
          MAXI[chan1]=data[chan1][temp-1];

          /* find peak of the activation in the reference channel */
          if ((temp<=851)&&(chan1==0)) peak[REFCH]=temp-1;
        }
      }
    }
    csp=0;      /* reset the current sample */
}


/******************/

void PeakRef()

/******************/

/* This routine detects the activation at the reference channel */
{
  float tempa,tempb,tempc;
  int posa,posb,posc,posa,posb,posc,nega,negb,negc;

  /* positive slope at the maximum amplitude */
  posa=FindSlop(REFCH,peak[REFCH]);
  posb=POSLP;  /* maximum positive slope */
  /* positive slope right before the maximum negative slope */
  posc=FindSlop(REFCH,WHNEG,1);

  nega=peak[REFCH];    /* the maximum amplitude */
```

```
  /* the amplitude after the maximum positive slope ends */
  pkb=WHPOS;
  /* the amplitude before the maximum negative slope starts */
  pkc=WHNEG;

  /* negative slope at the maximum amplitude */
  nega=FindSlop(REFCH,peak[REFCH],-1);
  /* negative slope right after the maximum positive slope */
  negb=FindSlop(REFCH,WHPOS,-1);
  negc=NESLP;    /* maximum nengative slope */

  /* if the maximum amplitude and the maximum positive slope and */
  /* the maximum negative slope do not occur at the same point */

  if ((pka!=pkb)&&(pkb!=pkc)&&(data[REFCH][pka]!=0)&&(posb!=0)&&(negc!=0))
  {
    /* weighting factor at the maximum amplitude */
    tempa=1 + (posa/posb) + (nega/negc);
    /* weighting factore at the maximum positive slope */
    tempb=(data[REFCH][pkb]/data[REFCH][pka])+1+(negb/negc);
    /* weighting factor at the maximum negative slope */
    tempc=(data[REFCH][pkc]/data[REFCH][pka])+(posc/posb)+1;

    /* if the weighting factor at the maximum positive slope is largest */
    /* then the activation is where the maximum positive slope occurs   */
    if ((tempb>tempa)&&(tempb>tempc)) peak[REFCH]=pkb;

    /* if the weighting factor at the maximum negative slope is largest */
    /* then the activation is where the maximum negative slope occurs   */
    if ((tempc>tempa)&&(tempc>tempb)) peak[REFCH]=pkc;
  }
}


/******************/

int FindSlop(ch,pk,sign)
int ch,pk,sign;

/******************/

/* This routine finds the slope at a segment which can be either up slope */
/* or down slope.                                                         */
{
  int sip;
  int sampl,diff;
  sip=0.0;
  diff=1;
  sampl=pk;

  while(diff>0)
  {  diff=data[ch][sampl]-data[ch][sampl-sign];
     sampl=sampl-sign;
  }
  if (pk-sampl-sign!=0)
  sip=(data[ch][pk]-data[ch][sampl-sign]) / (pk-sampl-sign);
```

```
  return sig;
}


/*******************/

int FindPeak(int ch,int lb,int ub,int pky)

/*******************/

/* This routine finds the peak of the activation at a specific channel */
/* in a certain range which is specified by lower and upper bound.     */
{
  int sampl,peakx;
  peakx = lb;      /* initialize the peak at the lower bound */

  for (sampl=lb;sampl<ub;sampl++)
    if (data[ch][sampl] > pky)
    {  pky = data[ch][sampl];
       peakx=sampl;
    }

  return(peakx);
}

:
/*******************/

void EnbInt()

/*******************/

/* This routine sets the vector interupt so that whenever counter 0 timed */
/* out after every millisecond interupt #3 will cause a jump to the       */
/* sampling interupt routine.                                             */
{
   unsigned int  vect;

   /* set the vector interupt to the sampling interupt routine */
   setvect(VECINT+INT, sampling);

   disable();        /* disable all the interupts */

   vect=inport(MASKP);            /* read current vector interupt */
   vect=vect&(~(1<<INT)|0x00ff);  /* add interupt #3 */
   outport(MASKP,vect);           /* write new vector interupt */

   /* counter 0 will be reset after all the processes enable again */
   SetTimer(0xff,0x00);

   /* enable all the processes */
   enable();
}


/*******************/

void DisInt()
```

```
xxxxxxxxxxxxxxxxx/

/* This routine  disables interupt #3 after 1.2 second of data have */
/* been collected at all the channels.                              */

{
   unsigned int vect;

   disable();    /* disable all the processes */

   vect=inport(MASKP);   /* read current vector interupt */
   vect=vect|(1<<INT);   /* mask out interupt #3 */
   outport(MASKP,vect);  /* write new vector interupt */

   enable();     /* reable all the processes */
}


/xxxxxxxxxxxxxxxxx/

void SeTimer(b1,b2)
unsigned int b1,b2;

/xxxxxxxxxxxxxxxxx/

/* This routine loads counter 0 with a certain amount of time */
/* which is specified by two variables b1 and b2              */

{
  outportb(TIME0,b1);
  outportb(TIME0,b2);
}


/xxxxxxxxxxxxxxxxx/

void search()

/xxxxxxxxxxxxxxxxx/

/* The functions of this routine:                               */
/*     - To find the maximum amplitude at a specific channel    */
/*     - To find the maximum positive slope                     */
/*     - To find the maximum negative slope                     */
/*     - To find the activation                                 */

{
  int  ch,sampl,pka,pkb,pkc,posa,posb,posc,nega,negb,negc;
  float tempa,tempb,tempc;

  for (ch=1; ch<MAXCH; ch++)
  {
    /* After the activation of the reference channel, find the peak      */
    /* of the other channels in an interst window in which the activation */
    /* should occurs                                                     */

    /* the maximum amplitude */
```

```
    pka=FindPeak(ch,peak[REFCH]-80,peak[REFCH]+349,0);
    pkb=FindPeak(ch,peak[REFCH]+80,peak[REFCH]+349);
    pkc=WHPOS;  /* the amplitude at the maximum positive slope */
    pkc=WHNEG;  /* the amplitude at the maximum negative slope */

    posa=FindSlop(ch,pka,1); /* the positive slope at the maximum amplitude */
    posb=FindSlop(ch,pkb,1); /* the maximu positive slope */
    /* the positive slope before the maximum negative slope */
    posc=FindSlop(ch,pkc,1);

    nega=FindSlop(ch,pka,-1); /* the negative slope at the maximum amplitude */
    /* the negative slope after the maximum positive slope */
    negb=FindSlop(ch,pkb,-1);
    negc=FindSlop(ch,pkc,-1); /* the maximum negative slope */

    /* initialize the activation where the maximum amplitude occurs */
    peak[ch]=pka;

    /* if the maximum amplitude , maximum positive slope and */
    /* maximum negative slope do not occur at the same place */
    if ((pka!=pkb)&&(pkb!=pkc)&&(data[ch][pka]!=0)&&(posb!=0)&&(negc!=0))
    {
      /* the weighting factor at the maximum amplitude */
      tempa=1 + (posa/posb) + (nega/negc);
      /* weighting factor at the maximum positive slope */
      tempb=(data[ch][pkb]/data[ch][pka])+1+(negb/negc);
      /* weighting factor at the maximum negative slope */
      tempc=(data[ch][pkc]/data[ch][pka])+(posc/posb)+1;

      /* if the weighting factor at the maximum positive slope is largest */
      /* then activation is where the maximu  amplitude occurs            */
      if ((tempb>tempa)&&(tempb>tempc)) peak[ch]=pkb;

      /* if the weighting factor at the maximum neagtive slope is largest */
      /* then the activation is where the maximum negative slope occurs   */
      if ((tempc>tempa)&&(tempc>tempb)) peak[ch]=pkc;
    }
  }
}


/*******************/

void slpwdow(int ch,int lb,int ub)

/*******************/

/* This routine finds the maximum positive and negative slope at a */
/* specific channel in a window which is specified by two          */
/* variables lb and ub */

{  int up,botpos,topneg,temp,slp;

   /* initialization */
   up=0;
   WHPOS=WHNEG=lb;
   topneg=lb;
   botpos=temp=lb;
   POSLP=NEGLP=0;
```

```
   while (temp<sub)
   {
      /* down slope */
      if (data[ch][temp+1]<=data[ch][temp])
      {  slp=0;
         if ((up==1)&&(temp!=botpos)) /* if down slope right after up slope */
         {   slp=(data[ch][temp]-data[ch][botpos])/(temp-botpos);
             if (slp>POSLP) { POSLP=slp; WHPOS=temp; }
         }
         botpos=temp+1;
         up=0;
      }
      /* up slope */
      if (data[ch][temp+1]>data[ch][temp])
      {  slp=0;
         if ((up==0)&&(temp!=topneg)) /* if up slope right after down slope */
         {  slp=(data[ch][temp]-data[ch][topneg])/(temp-topneg);
            if (slp<NESLP) { NESLP=slp; WHNEG=topneg; }
         }
         topneg=temp+1;
         up=1;
      }
      temp+=1;
   }
}


/*******************/

void message(msg)

/*******************/

/* This routine sends a message to the screen and turns on the speaker */
/* whenever a wrong key is pressed.                                    */

{
   setcolor(4);
   outtextxy(130,462,msg);  /* send a message */

   sound(440);   /* turn on the speaker */
   delay(100);
   nosound();    /* turn off the speaker */
}


/*******************/

void clrmsg(x1,y1,x2,y2)
int x1,x2,y1,y2;

/*******************/

/* This routine clears the message */
{
   setfillstyle(SOLID_FILL,7);
   bar(x1,y1,x2,y2);
}
```

```
/*******************/

void interrupt sampling()

/*******************/

/* This routine reads the digitized signals at all the channels */
/* every time the interupt # 3 is generated or in other word    */
/* the counter 0 is timed out.                                  */
{
   int ch,i;

   disable(); /* disable all the processes */

   for (ch=0;ch<3;ch++)
      data[ch][csp]=ADConvert(ch);    /* read data at the A/D board */

   csp+=1;    /* increment the current sample */

   if (csp<=MAXSP) SeTimer(0x90,0x10);  /* reload counter 0 */

   else  DisInt();         /* disable inetrupt #3 if 1.2 second of data */
                           /* have been collected                       */
   outportb( CT8259,0x20 );
   enable();               /* enable all the processes again */
}
```

APPENDIX B

MAPPING PROGRAM LISTING

```
/*****************************************************************************
**                                                                          **
**      Name of module: MAPPING                                             **
**                                                                          **
**      This module displays the activation time at each location           **
**      and updates the map of the ventricles on the screen so              **
**      the surgeons and the cardiologist can see which location            **
**      is early or late.                                                   **
**      It can also store the digitized waveforms on the hard disk          **
**      or modify the activation time.                                      **
**                                                                          **
**      Written by : Thuthuy Bach                                           **
**                                                                          **
**      Starting date: May 1st 1990                                         **
**      Closing date:  Dec 24  1990                                         **
**                                                                          **
*****************************************************************************/


#include "c:mod2.c"     /* include activation time module */
#define fn2 "c:res.dat" /* data file to store the result of the operation */

int GDriver, /* graphic driver */
    GMode,   /* graphic mode */
    ckey,    /* current location on the ventricles */
    clor,
    mark;    /* sample where the activation occurs */
;
/* colors that represent the range of the activation time */
int label[6]={11,10,15,14,13,12};
int rul[6];         /* the ranges of the activation time */
int scale[MAXCH];   /* the autonomic gain at each channel to display on the screen */
box pos[20];        /* the menu */
FILE *res;          /* result data file */

void main()         /* main program */
{
  foreword();       /* the introduction of the the mapping module */
  InitMap();        /* initialization of the map */
  InitAD();         /* initialization of the A/D board */
  DisplayMap();     /* display the initial map */
  menu();           /* display the menu */
  InitFile();       /* Initialize the data files */
  keyin();          /* read the key board */

  while(ckey!=19)   /* while is not the ESC key */
  {
    if ( (!kbhit()) && (ckey<17) && (dreq==0) ) upsig(); /* collect signals */
    if (kbhit())  keyin();                               /* read key board */
  }

  fclose(dat);   /* close the data file */
  fclose(res);   /* close the result file */
  closegraph();  /* close the graph */
 }


 /********************/
```

```
void foreword()

/*****************/

/* This program introduces the mapping system to the user */
{
  int answer;

  clrscr();

  printf("\n"); printf("\n"); printf("\n");
  printf("               WELCOME TO THE MAP OF ACTIVATION TIME\n");
  printf("                         VERSION THUY 1.0\n");
  printf("\n");
  printf("           THIS PROGRAM WILL DISPLAY THE ACTIVATION TIME\n");
  printf("               AT DIFFERENT LOCATIONS OF THE HEART\n");
  printf("\n");
  printf("\n");
  printf("     Please press Esc key to stop or any other key to continue\n");

  while (kbhit()) getch();
  while (!kbhit());
  answer = getch();
  if (answer==27)  exit(0);
}
  :
/*******************/

void InitMap()

/*******************/

/* The functions of this routine:                              */
/*      - To dectect the graphic driver and graphic mode.      */
/*      - To initialize the graphic environment                */
/*      - To select the VGA graphic.                           */
/*      - To set up the menu of intructions.                   */
/*      - To reset the activation time at all location to zero.*/

{
  int i,j;

  detectgraph(&GDriver,&GMode);
  initgraph(&GDriver,&GMode,"");
  restorecrtmode();
  if (GDriver<9) { printf ("VGA graphics required\n"); exit(1); }

  ckey = clet = 0;
  pos[0].let = 'A';
  pos[1].let = 'B';
  pos[2].let = 'C';
  pos[3].let = 'D';
  pos[4].let = 'E';
  pos[5].let = 'F';
  pos[6].let = 'G';
  pos[7].let = 'H';
  pos[8].let = 'I';
```

```
  pos[9].let = "J";
  pos[10].let = "K";
  pos[11].let = "L";
  pos[12].let = "M";
  pos[13].let = "N";
  pos[14].let = "O";
  pos[15].let = "P";
  pos[16].let = "Q";
  pos[17].let = "ReM";
  pos[18].let = "See";
  pos[19].let = "Esc";

  for (i=0;i<=19;i++)
  {  pos[i].xm = 536;
     pos[i].ym = 14 + (21*i);
     pos[i].ti = 0;
  }
}


/********************/

void DisplayMap()

/********************/
:
/* The function of this routine is to display the initial map */
{
  int i;

  /* The X and Y coordinates of each point on the map */
  int poly1[66]={77,308,57,266,46,217,52,167,91,125,130,84,165,52,208,17,220,29,
                 232,41,248,45,262,46,266,48,309,50,353,60,397,81,430,102,465,
                 146,484,185,498,230,495,269,460,273,458,238,445,202,428,172,
                 399,139,374,119,336,105,299,98,263,98,190,64,89,182,110,297};
  int poly2[48]={460,273,458,238,445,202,428,172,399,139,374,119,336,105,299,98,
                 263,98,190,64,89,182,110,297,154,284,145,206,193,115,260,146,288,
                 147,316,154,344,165,364,175,386,200,400,223,411,251,417,279};
  int poly3[48]={154,284,145,206,193,115,260,146,288,147,316,154,344,165,364,175,
                 386,200,400,223,411,251,417,279,361,284,358,263,351,242,343,230,
                 329,214,316,207,297,200,277,196,257,195,210,178,196,225,204,266};
  int poly4[42]={361,284,358,263,351,242,343,230,329,214,316,207,297,200,277,196,
                 257,195,210,178,196,225,204,266,255,249,267,248,277,249,290,251,
                 295,252,302,256,305,263,306,276,305,290};
  int dspl[17][2]={ {220,18},{165,39},{130,70},{85,112},{50,154},{35,203},{45,252},
                    {66,301},{256,35},{291,39},{340,49},{389,70},{423,87},{460,126},
                    {483,168},{501,217},{500,266} };

  setgraphmode(GMode);
  cleardevice();
  clrscr();

  setcolor(label[4]);
  setfillstyle(SOLID_FILL,label[4]);
  fillpoly(33,poly1);
  setcolor(label[3]);
  setfillstyle(SOLID_FILL,label[3]);
```

```
  fillpoly(24,poly2);
  setcolor(label[2]);
  setfillstyle(SOLID_FILL,label[2]);
  fillpoly(24,poly3);
  setcolor(label[1]);
  setfillstyle(SOLID_FILL,label[1]);
  fillpoly(21,poly4);

  setfillstyle(SOLID_FILL,7);
  bar(0,399,640,480);          /* extent the screen */

  setcolor(9);
  for (i=0;i<=16;i++)
  outtextxy(dspl[i][0],dspl[i][1],pos[i].let);  /* display the locations */
}


/*********************/

void menu()


/*********************/

/* This routine displays the menu and the windows for observing the */
/* the input waveforms.                                             */
{
  int i;

  /* display the menu */
  setcolor(5);
  outtextxy(pos[0].xm,pos[0].ym,pos[0].let);
  setcolor(1);
  for (i=0;i<=111;i+=37)
     line(529+i,7,529+i,427);
  for (i=0;i<=425;i+=21)
     line(529,7+i,640,7+i);
  for (i=1;i<=19;i++)
     outtextxy(pos[i].xm,pos[i].ym,pos[i].let);

  /* display the windows */
  for (i=329;i<=455;i+=126)
     line(35,i,515,i);
  for (i=35;i<516;i+=480)
     line(i,329,i,455);
  outtextxy(12,357,"REF");
  outtextxy(12,399,"MAP");
  outtextxy(12,441,"SYN");
}


/*********************/

void InitFile()

/*********************/

/* This routine opens the data file to store all the digital signals */
```

```
/* that are collected from the patient  and the result file to store  *
/* all the activation time at each location that have been detected. *
{

  creq=0;   /* reset data acquisition flag */

  if ((dat=fopen(fn,"ab"))==NULL)  /* open data file */
  {
    message("File is not open");
    exit(1);
  }

  if ((res=fopen(fn2,"at"))==NULL) /* open result file */
  {
    message("File is not open");
    exit(1);
  }

}


/**********************/

void upsig()

/**********************/
{
 /* The functions of this routine:                                  */
 /*       - To find the local activation which have been detected   */
 /*         from the activation detection module.                   */
 /*       - To display the activation time on the screen.           */
 /*       - To display the waveform at each channel with the        */
 /*         automatic gain in the windows.                          */
 /*       - To show where the activation takes place at each channel */
 /*         for verification or modification.                       */

 {

    int i,x,ch,y1,y2,top,bot;
    float temp;
    char *str;

    actime();   /* find the activation at each channel */
    pos[ckey].t1 = peak[1]-peak[0];     /* local activation time */
    pos[ckey].t2 = peak[2]-peak[1];

    clrmsg(36,330,514,454);    /* Clear the previous waveforms in the window */

    y1=8+21*ckey;               /* clear the previous activation time */
    y2=y1+18;                   /* in the menu */
    clrmsg(567,y1,602,y2);
    clrmsg(604,y1,639,y2);

    setcolor(15);      /* display the new activation time */
    outtextxy(pos[ckey].xm+33,pos[ckey].ym,itoa(pos[ckey].t1,str,10));
    outtextxy(pos[ckey].xm+70,pos[ckey].ym,itoa(pos[ckey].t2,str,10));

    /* find the automatic gain for each channel so that the waveform */
```

```
   * at each frame can be fitted and observed in the window      */
  for (ch=0;ch<3;ch++)
  {  temp= MAX[ch]-MINI[ch]/40.0;
     temp=log(temp)/log(2);
     scale[ch]=(int)temp+1;
     if(scale[ch]<0) scale[ch]=0;
  }

  for (x=37;x<=514;x++)  /* display the waveform of each channel */
  {
     if ( ((peak[0]+x-76)>=0) && ((peak[0]+x-75)<MAXSP-1) )
     {
        for(ch=0;ch<3;ch++)
        {  bot=370-(ch*42);
           top=bot-40;
           y1=bot-((data[ch][peak[0]+x-76]-MINI[ch]))>>scale[ch]);
           y2=bot-((data[ch][peak[0]+x-75]-MINI[ch]))>>scale[ch]);
           if ( (y1<=bot)&&(y2<=bot)&&(y1>=top)&&(y2>=top) )
              line(x-1,y1,x,y2);
        }
     }
  }

  setcolor(1);  /* display the vertical bar to show where the activation is detected */
  for (ch=0;ch<3;ch++)
     if ( ((peak[ch]>=peak[0])&&(peak[ch]-peak[0]<=439)) || ((peak[0]>=peak[ch])&&(peak[0]-peak[ch]<=381)) )
       line(75+peak[ch]-peak[0],330+42*ch,75+peak[ch]-peak[0],370+42*ch);

  clet=ckey;     /* current position is the current key has been pressed */
  mark=peak[0];  /* the position where the activation of the reference *
                 /* channel took place */
}


/********************/

void keyin()

/********************/

/* The funtion of this routine is to read the keyboard in order to perform */
/* a specific command for the user.                                         */

{
   int key;

   key = getch(); /* read the key in */

   /* if any letter from a to q is pressed the cursor in the menu is moved */
   /* to the corresponding location and reset the data acquisition flag */
   if ((key>=65)&&(key<=81)) { dreq=0; jump(key-65); }

   /* if any letter from A to Q is pressed the cursor in the menu is moved */
   /* to the corresponding location and reset the data acquisition flag */
   else if ((key>=97)&&(key<=113)) { dreq=0; jump(key-97); }

   /* if letter R is selected then remark the activation on any displayed */
```

```
   * select *
   else if (key==82||key==114) scr();

   /* if letter S is selected then update the map */
   else if (key==83||key==115) upmap();

   /* if the enter key is selected */
   else if (key==13)
      { if (ckey==19)  :

        /* and if the current cursor at the remark menu then remark the */
        /* activation time */
        else if (ckey==17)  scr();

        /* if the current cursor at the see menu then update the map */
        else if (ckey==18)  upmap();

        /* if the current cursor at any other position then store the */
        digital waveform at this position into the hard disk */
        else { dreq=-1;                 /* set the data acquisition flag */
               fputc(ciet+97,dat);      /* store the position */
               fwrite(data,2,3600,dat); }        /* store the waveform */
      }

   /* if one of the arrow key is pressed */
   else if (key==0)
       {  key=getch(); /* get the arrow key */
          move(key);   /* move the cursor */
          keyin();     /* read next key */
       }
   else if (key==27)  ckey=19;   /* if ESC key key is pressed */
   else      /* any other keys will generate an error message */
       {
          message("Undefined letter pressed!");
          jump(ckey);    /* move cursor to current key */
          keyin();       /* read next key */
       }
}

/*******************/

void upmap()

/*******************/

/* The functions of this routine                                    */
/*      - To find the maximum and minimum activation times and the */
/*        locations where they take place.                          */
/*      - To display a corresponding color for each segment of the */
/*        activation time.                                          */
/*      - To display a corresponding color for each recoprded      */
/*        location.                                                 */
/*      - To display the minimum activation time at a particular   */
/*        location in red.                                          */
/*      - Store the result into the hard disk.                     */
/*      - Reset the data acquisition flag.                         */

{
```

```
int pogon[][4]= {206, 43,193, 29,205, 17,220, 29},{190, 64,165, 52,193, 29,218, 43},
                {159,101,130, 84,165, 52,190, 64},{122,143, 91,125,130, 84,159,101},
                { 89,182, 52,167, 91,125,122,143},{ 96,224, 46,217, 52,167, 89,182},
                {103,263, 57,266, 46,217, 96,224},{110,297, 77,306, 57,266,103,263},
                {263, 98,238, 87,248, 45,266, 48},{299, 98,263, 98,266, 48,309, 50},
                {336,105,299, 98,309, 50,353, 60},{374,119,336,105,353, 60,397, 81},
                {399,139,374,119,397, 81,430,102},{428,172,399,139,430,102,465,146},
                {445,202,428,172,465,146,464,185},{458,238,445,202,464,185,498,230},
                {460,273,458,238,498,230,496,269} };

int i,j,min,max,minval,maxval,y1,y2,div,clr;
char *str;

/* Initialization */
maxval=1;
minval=1000;
max=min=-1;

/* find the maximum and minimum activation times and the locations where */
/* they take place */
for (i=0;i<=16;i++)
{  if(pos[i].t1>maxval) { maxval=pos[i].t1; max=i; }
   if( (pos[i].t1<minval) && (pos[i].t1>0) ) { minval=pos[i].t1; min=i; }
}

/* divide the range from the minimum to the maximum activation time */
/* into 6 segments */
if ((max!=-1)&&(min!=-1))
{
 div=(pos[max].t1-pos[min].t1)/6;
 j=0;
 for (i=5;i>=0;i--)
 {  rul[i]= pos[min].t1 + (j*div);
    j+=1;
 }

 CoRuler1(); /* display a corresponding color for each segment */
          /* of the activation time */

 /* Display a corresponding color for each recorded location */
 for (i=0;i<=16;i++)
 {
   if (pos[i].t1>0)       /* only if the location has been recorded */
   {
     for (j=5;j>=1;j--)  /* chose the corresponding color */
       if ( (pos[i].t1>=rul[j]) && (pos[i].t1<rul[j-1]) )
         clr=label[j];
     if (pos[i].t1 >= rul[0])  clr = label[0];

     setcolor(clr);               /* color this location */
     setfillstyle(SOLID_FILL,clr);
     fillpoly(4,pogon[i]);

     if (i==min) setcolor(12);    /* display the minimum activation */
     else setcolor(15);           /* time at a particular location  */
     y1=8+21*i;                   /* in red so that the surgeons    */
     y2=y1+18;                    /* can notice right away.         */
     clrmsg(567,y1,602,y2);
     outtextxy(pos[i].xm+35,pos[i].ym,itoa(pos[i].t1,str,10));
```

```
        /* store the result in the result file */
        fprintf(res,"%s       %3d     %3d\n", pos[i].let,pos[i].x1,pos[i].x2);
      }
    }
    crec=-1;  /* reset data acquisition flag */
  }
}


/**********************/

void CoRuler()

/**********************/

/* The functions of this routine are to clear the previous color scale */
/* and display the new color scale.                                    */
{
  int i;
  char *str;

  clrmsg(0,0,22,250);  /* clear the previous color scale */

  for (i=0;i<6;i++)    /* Display the new color scale */
  {
    setfillstyle(SOLID_FILL,label[i]);
    bar(0,42*i,20,20+42*i);                  /* show the color */
    setcolor(label[i]);
    outtextxy(0,31+42*i,itoa(ru[i],str,10));  /* show the activation time */
  }
}


/**********************/

void scr()

/**********************/

/* The functions of this routine is to allow the user move the vertical bar  */
/* at each channel to a new position to change the activation time.          */

{
   int key1,key2,y1,y2;
   char *str;

   int w=0;    /* initialize the cursor to the vertical bar of the first window */

   setcolor(1);  /* remove the cursor from the current position in the menu */
   outtextxy(pos[17].xm,pos[17].ym,pos[17].let);
   outtextxy(pos[ckey].xm,pos[ckey].ym,pos[ckey].let);

   setcolor(5);  /* move the cursor to the current recording location */
   outtextxy(pos[clet].xm,pos[clet].ym,pos[clet].let);

   window (5,w);

   key1=getch(); /* read the key board */
```

```
   while ((key1!=82)&&(key1!=27))  /* while the key returning to the main menu */
                                   /* have not been pressed */
   {
      if (key1==0)                 /* if the arrow key is pressed */
      {  key2=getch();             /* read the arrow key */

         if ((key2==72)&&(w!=0))          /* move the cursor down to the next channel */
           {wdow(1,w); w=wdow(5,w-1);}
         else if ((key2==80)&&(w!=2))     /* move the cursor up to the next window */
           {wdow(1,w); w=wdow(5,w+1);}
         else if (key2==75) cursor(-1,w); /* move the cursor to the left at the same channel */
         else if (key2==77) cursor(1,w);  /* move the cursor to the right at the same channel */
      }
      else message("Use arrow key or type R");  /* else send the warning message */
      key1=getch();                             /* read key board again */
   }

   clrmsg(105,462,410,472); /* clear the message */
   ckey=clet;               /* set the cursor to the current recording location */

   /* store the new result in hard disk */
   fprintf(res,"%s       %3d     %3d   Cor\n",pos[clet].let,peak[1]-peak[0],peak[2]-peak[1]);

   w=wdow(1,w); /* remove the cursor from the window */

   pos[ckey].t1 = peak[1]-peak[0];  /* find the new activation time */
   pos[ckey].t2 = peak[2]-peak[1];

   y1=8+21*ckey;          /* clear the previous activation time */
   y2=y1+18;
   clrmsg(567,y1,602,y2);
   clrmsg(604,y1,639,y2);

   setcolor(15);          /* display the new activation time */
   outtextxy(pos[ckey].xm+33,pos[ckey].ym,itoa(pos[ckey].t1,str,10));
   outtextxy(pos[ckey].xm+70,pos[ckey].ym,itoa(pos[ckey].t2,str,10));

   dreq=-1; /* reset the data acquisition flag */
}


/*******************/

void cursor(int dir,int w)

/*******************/

/* The function of this routine is to move the vertical bar at a channel to */
/* the left or to the right.                                                 */

{
   int x,y1,y2,bot,top;

   clrmsg(105,462,410,472); /* clear the message */

   if ((dir>0)&&((peak[w]-mark)<=439)) ;     /* prevent the vertical bar move out of the */
   else if ((dir<0)&&((mark-peak[w]))<=38)) ; /* window either on the left or on the right */
```

```
   else
   {  x=75-peak[w]-mark; /* the sample on the current channel on which the cursor *.
                         /* is positioning */
      if ( ( (dir<0)&&(mark+x-75>1) ) || ( (dir>0)&&(mark+x-75<MAXSP-2) ) )
      {
        wdow(7,w);     /* turn off the vertical bar at the previous sampl */
        peak[w]+=dir;  /* move the avtivation time to the next sample */
        wdow(5,w);     /* move the vertical bar at the next sample */

        bot=370+(w*42);  /* redisplay the previous sample of the waveform */
        top=bot-40;
        y1=bot-((data[w][mark+x-dir-75]-MINI[w]))>>scale[w]);
        y2=bot-((data[w][mark+x-75]-MINI[w]))>>scale[w]);
        setcolor(15);
        if ( (y1<=bot)&&(y2<=bot)&&(y1>=top)&&(y2>=top) )
           line(x-dir,y1,x,y2);
      }
   }
}


/*******************/

int wdow(int cl,int w)

/*******************/
{
/* The function of this routine is to turn on or turn off the vertical bar at */
/* a specific channel.                                                         */

{
   int bot,top;

   clrmsg(105,462,410,472); /* clear pevious message */

   /* display the vertical bar only when its position is still inside the window */
   if ( ((peak[w]>=mark)&&(peak[w]-mark<=439)) || ((mark>=peak[w])&&(mark-peak[w]<=36)) )
   {
      setcolor(cl);
      bot=370+42*w;
      top=bot-40;
      line(75+peak[w]-mark,top,75+peak[w]-mark,bot);
      return(w);
   }
}


/*******************/

void move(k)
int k;

/*******************/

/* The function of this procedure is to move the cursor up or down */
/* to the desired location in the menu.                            */
{
  switch (k)
  {
```

```
    case 72:  /* up arrow key */
    { if (ckey==0) { jump(18);break; }  /* move to the last location */
      else { jump(ckey-1);break; }      /* move up to next location  */
    }
    case 80:  /* down arrow key */
    { if (ckey==18) { jump(0);break; }    /* move to the first location */
      else { jump(ckey+1);break; }        /* move down to next location */
    }
    default : message("Undefined key pressed");
  }
}


/***************/

void jump(k)
int  k;

/***************/

/* This procedure removes the cursor from the previous location and  */
/* might light the current location where the cursor is.             */

{
  clrmsg(105,462,410,472); /* clear previous message */
  /* remove the cursor at the previous location */
  setcolor(1);
  outtextxy(pos[ckey].xm,pos[ckey].ym,pos[ckey].let);

  ckey=k;     /* get the current key */

  /* move the cursor to the current location */
  setcolor(5);
  outtextxy(pos[ckey].xm,pos[ckey].ym,pos[ckey].let);
}
```

APPENDIX C

DATA-RETRIEVAL PROGRAM LISTING

```
/**************************************************************************
**                                                                       **
**  Name of module: DSPL.H                                               **
**                                                                       **
**  This module contains the header and the global declarations          **
**  for the module DSPL.C                                                **
**                                                                       **
**  Written by: Thuthuy Bach                                             **
**                                                                       **
**  Starting date: May 10 1990                                           **
**                                                                       **
**  Closing date:  Nov 15 1990                                           **
**                                                                       **
**************************************************************************/

/* The following libraries are included */
#include <dos.h>
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#include <math.h>
#include <process.h>

#define fn "c:test.dat"   /* File recorded during the operations */


void main(void);          /* Main program */
void InitMap(void);       /* Initialize the map */
void InitFile(void);      /* Initialize the file */
void display(void);       /* Display the signals */
void keyin(void);         /* Read the request */
void NewSetint();         /* Dispaly new set of data */
```

```
/*****************************************************************************
**                                                                          **
**   Module name: DSPL.C                                                    **
**                                                                          **
**   Function:    This module retrieve the data that recorded               **
**                during the operation for post-operation study.            **
**                                                                          **
**   Written by:  THUTHUY BACH                                              **
**                                                                          **
**   Starting date: May 10 1990                                             **
**                                                                          **
**   Closing date:  Nov 15 1990                                             **
**                                                                          **
*****************************************************************************/

#include "c:dspl.h" /* include the header module */

int GDriver, /* Graphics driver number */
    GMode,   /* Graphics mode */
    ckey,    /* current key */
    setnum,  /* set number */
    offset;  /* sample number in a set */

int data[3][1200]; /* buffer of data */

FILE *dat; /* Data file */

/***************/

void main()

/***************/

{
  InitMap();        /* Initialize the map */
  InitFile();       /* Initialize the data file */

  while(ckey!=27)   /* while current request is not ESC */
  {
    if (kbhit())    /* If there is a request */
    { keyin();      /* Read the request */
      display();    /* Display the corresponding signals */
    }
  }
  fclose(dat);      /* Close the data file */
  closegraph();     /* Close the graphics environment */
}

/***************/

void InitMap()

/***************/

/* The function of this procedure is to initialize the graphics */
/* environment, the set number and the sample number in a set   */

{
  detectgraph(&GDriver,&GMode); /* Get the graphics driver and mode */
```

```
  initgraph(&GDriver,&GMode,""); /* Initialize the graphics environment */
                                 /* the graphics driver and mode number */
  restorecrtmode();              /* Restore CRT mode */
  setgraphmode(GMode);           /* Set the graphics mode */
  cleardevice();                 /* Clear the device */
  setnum=0;                      /* Reset the set number to the first set */
  offset=0;                      /* Reset the sample number the first     */
                                 /* in the set */
}

/***************/

void InitFile()

/***************/

/* The functions this procedure are                        */
/*     - To open the data file.                            */
/*     - To initialize the pointer to the beginning of     */
/*       the file.                                         */
/*     - To display the new set of the file.               */

{
  if ((dat=fopen(fn,"rb"))==NULL) /* open the data file */
  {
    printf("File is not open");
    exit(1);
  }
  ckey=0;                 /* Reset current request */
  fseek(dat,0L,SEEK_SET); /* Reset the pointer to the beginning */
                          /* of the file */
  NewSet(1);              /* Read the first set in the file */
  display();              /* Display the first set */
}

/***************/

void display()

/***************/

/* The functions of this procedure are:                          */
/*    - To clear the screen .                                    */
/*    - To display the setnumber, sample number and the signals */

{
   float temp;   /* scale of display */
   int y1,y2,    /* the potential of the signals */
       x,x1,     /* the smple number */
       bot,top,  /* the top and bottom of each window on the screen */
       ch;       /* channel number */
   char *str;    /* string of message */

   temp=155.0/4096.0;   /* Calculate the scale of the screen */
   clrscr();            /* clear the screen */

   setcolor(7);         /* set the color of the back plane */
   setfillstyle(SOLID_FILL,7);  /* extent the screen */
```

```
 bar(0,399,640,480);

 setcolor(15);                /* set white color */
 line(0,156,600,156);         /* divide the screen into three windows */
 line(0,312,600,312);
 line(0,468,600,468);

 setcolor(14);                /* set yellow color */
 outtextxy(0,460,itoa(setnum,str,10)); /* display the setnumber */
 outtextxy(0,470,itoa(offset,str,10)); /* display the sample number */

 for (x=0;x<=640;x++)                  /* display the signals */
 {
    x1=x + offset; /* Find current sample to display */
    if (x1<1199)   /* if current sample not the last sample in the buffer */
    {
       for(ch=0;ch<3;ch++)   /* Display the waveform of each channel */
       {  bot=155+(ch*156);  /* the bottom line of the windoow */
          top=bot-155;       /* the top line of the window */
          y1=bot-(int)(((data[ch][x1]*30.0)+2048.0)*temp);
          y2=bot-(int)(((data[ch][x1+1]*30.0)+2048.0)*temp);
          if ( (y1<=bot)&&(y2<=bot)&&(y1>=top)&&(y2>=top) )
              /* draw the straight line between two samples */
              line(x,y1,x+1,y2);
       }
    }
  }
}

/***************/

void keyin()

/***************/

/* This procedure reads the key board to receive the request from */
/* the user then calls other subroutines to perform the request.  */

{
   ckey = getch();   /* Get the request */
   if (ckey!=27)     /* if the request is not ESC */
   {
     /* display the next set */
     if (((ckey==102)||(ckey==70))&&(setnum<26)) NewSet(1);
     /* display the previous set */
     if (((ckey==98)||(ckey==66))&&(setnum>1)) NewSet(-1);
     /* display the next cycle in the same set */
     if (((ckey==110)||(ckey==78))&&(offset<2340)) offset+=640;
     /* display the previous cycle in the same set */
     if (((ckey==98)||(ckey==66))&&(offset>640)) offset-=640;
   }
}

/***************/

void NewSet(int num)

/***************/
```

```
/* This procedure reads the next set of data from the data file */
/* into the buffer.                                             */
{
   int dum;
   setnum+=num;     /* calculate current set number */
   offset=0;        /* set the current sample to the first sample */
                    /* in the set */
   if(num==-1)      /* if previous set is required */
     /* set the pointer to the beginning of previous set */
     fseek(dat,-7202,SEEK_CUR);

   /* get the name of location where the set were recorded */
   dum=fgetc(dat);
   fread(data,2,3600,dat);  /* read the data oof the set into the buffer */
}
```

APPENDIX D

HARDWARE COMPONENTS OF THE SYSTEM

The hardware components of the system, as shown in Figure D.1, include:

- Two 9-pin male connectors. (206705-1 AMP)
- Two 9-pin female connectors. (206708-1 AMP)
- One 40-pin male connector.
- Three 1-pole, 6-position rotary switches. (71BD36-01-1-AJN Grayhill)
- Three 1-pole, 10-position rotary switches. (71BD36-01-1-AJN Grayhill)
- One on/off switch.

*** Two 9-pin male connectors:**

These connectors receive the input signals from 6 electrodes that are recorded from the patient. Before the signals are manipulated by the system, they are passed to the two 9-pin female connectors so that other equipment in the lab can also access these signals.

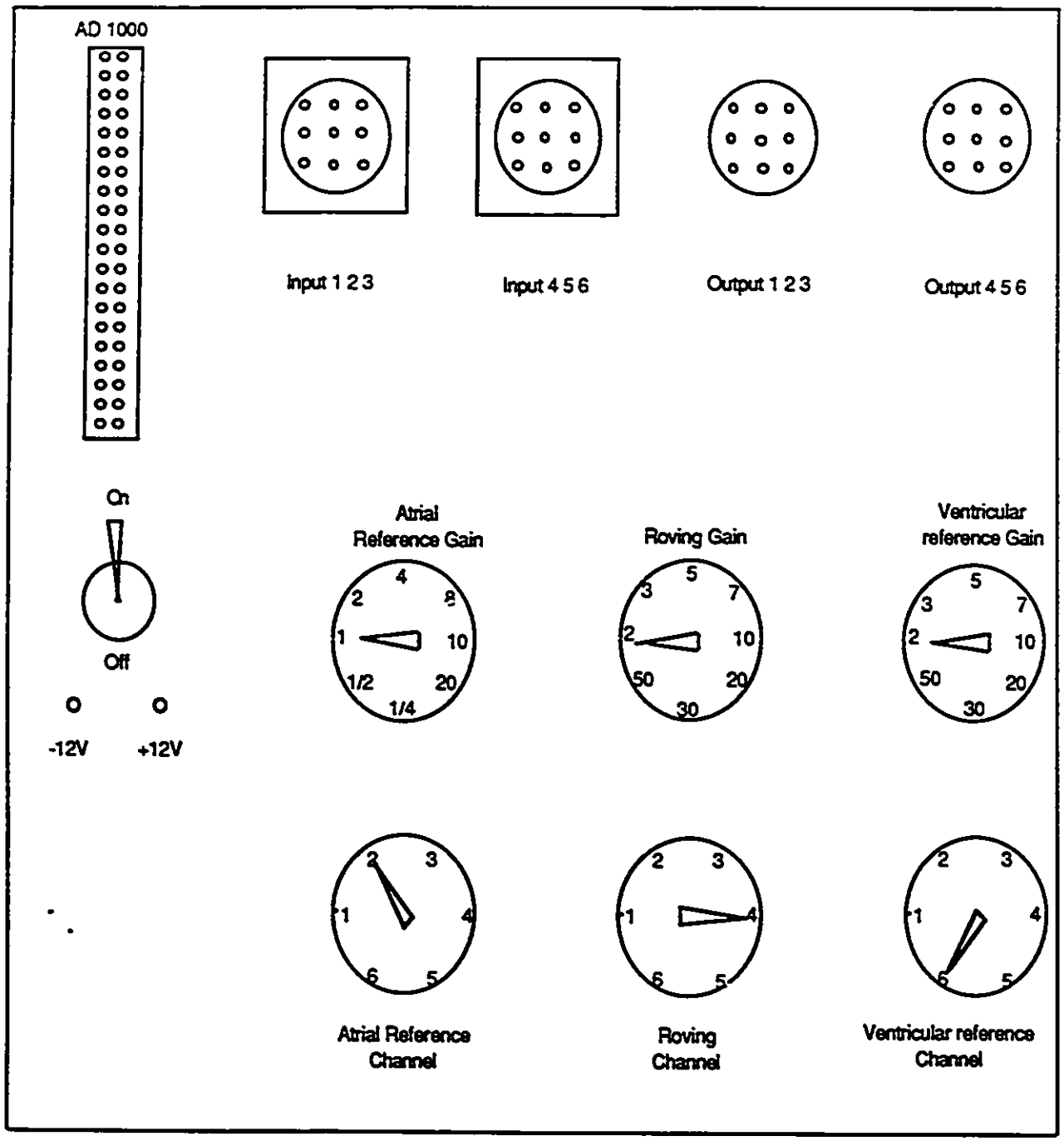

Figure D.1 Hardware configuration of the system.

* **Two 9-pin female connectors:**

These connectors deliver the signals from the patient to other equipment in the lab.

* **One 40-pin male connector:**

After the signals are buffered, filtered and amplified, they go to the A/D board through this 40-pin connector:

- Pin 6 is connected to the signal of atrial reference channel.
- Pin 25 is connected to the signal of the roving electrode.
- Pin 5 is connected to the ventricular reference channel.
- Pin 1 and pin 4 are connected to the isolated ground of the signals.
- Pin 19 is to provide +12 V to the op-amps.
- Pin 20 is to connected to the -12 V to the op-amps.
- Pin 16 (End of Conversion bit) is connected to pin 31 (port B) since this port is not available at the I/O connector. Port B must be configured as mode 0 input so that when this port is read, the most significant bit will indicate the conversion status: EOC will be a logic low while the conversion is in progress, and will go high and remain high after a conversion is completed [28].

* **Three 1-pole 6-position rotary switches:**

- RS1 is used to select one of 6 channels to be the atrial reference channel.
- RS2 is used to select one of 5 other channels to connect to the roving electrode.
- RS3 is used to select one of 4 remaining channels to connect to the ventricular reference channel.

* **Three 1-pole 10-position rotary switches:**

Fixed gain amplification is setting one level of gain for all the amplifiers. Variable gain amplification refers to the ability to change the gain of each amplifier individually and independently of all amplifiers.

* **One on/off switch:**

This switch is used to supply the +12 V and -12 V from the computer to the system.

APPENDIX E

MAN-MACHINE INTERFACE

This appendix presents the interface between the user and the computerized cardiac mapping system.

Hardware:

First, the input from channel 1, 2, 3 should be connected to the first 9-pin male connector and the input from channel 4, 5, 6 are connected to the second 9-pin male connector. Any other equipment in the operating room can access the signal from channel 1, 2, 3 or from channel 4, 5, 6 through the first or the second 9-pin female connector respectively. (See Figure D.1).

The upper left rotary switch is used to obtain the gain at the reference channel so that the amplitude of the signals from this channel is at the suitable level. The minimum or maximum amplitude of the signal should not be smaller than -5 V or greater than +5 V otherwise there would be cut off at the waveform and the activation time cannot be detected precisely because the A/D board can convert only the signals from -5 V to +5 V. However, when the maximum amplitude of the signal is very small, the signal might not be seen. Therefore, it depends on the amplitude of the original signals to select one variable gain in the range from 1/4 to 20. The amplitude of the original signal at this channel is usually very large (from -15 V to +15 V) since this is where the pacing stimulus is applied. Thus, the attenuation (gain of less than one unit) would be useful.

The lower left rotary switch can be used to select one of the 6 available channels to be the reference channel.

The upper middle rotary switch is used to obtain the gain at the roving channel so that the amplitude of the signals from this channel is at the suitable level. The amplitude of the signals at this channel is usually much smaller the that of the reference channel; Thus, the attenuation is not necessary at this channel. The gain at this channel can vary from 2 to 50.

The lower middle rotary switch can be used to select one of the remained channels to be the roving channel.

The upper right rotary switch is used to obtain the gain at the second reference channel so that the amplitude of the signals from this channel is at the suitable level. The amplitude of this channel is also much smaller than that of the reference channel. The gain at this channel can vary from 2 to 50.

The lower right rotary switch is used to select the remained channel to be the second channel.

The on/off switch should be at the on position to supply the power to the system.

Software :

The program is written in C language and the executable file of this program is called "MAP"". After the program is evoked, the introduction screen will appeared as shown in Figure E.1.

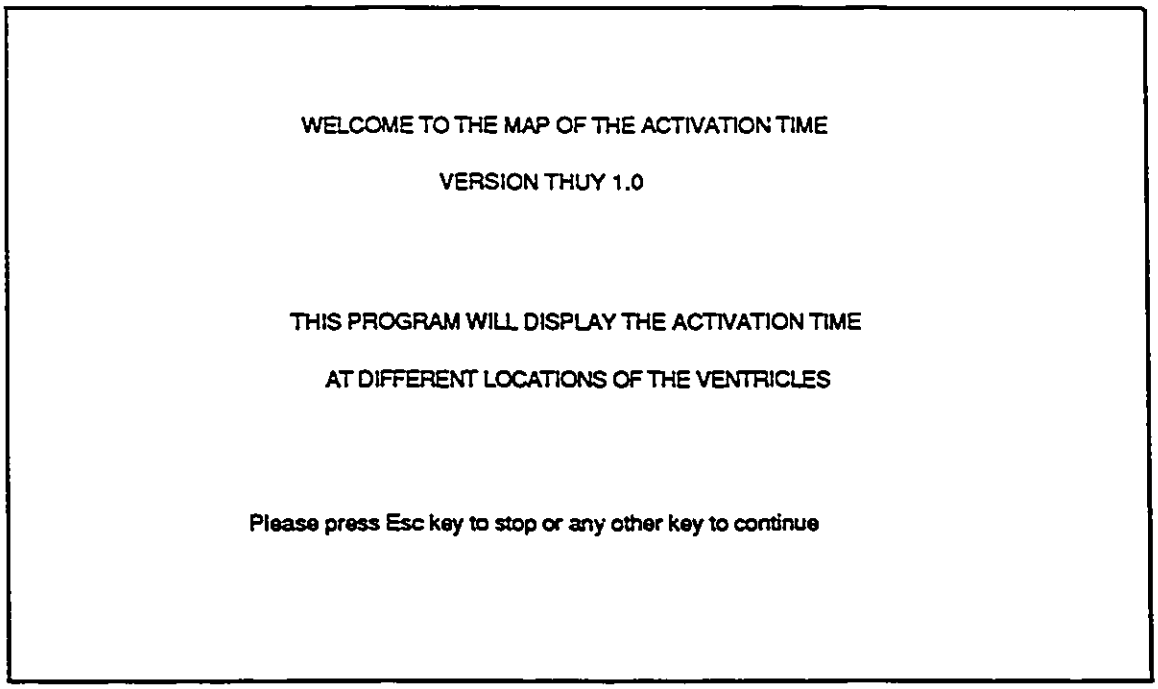

Figure E.1 Introduction Screen.

From this screen, if any other key (except ESC key) is pressed, the initial screen will be displayed as shown in Figure E.2.

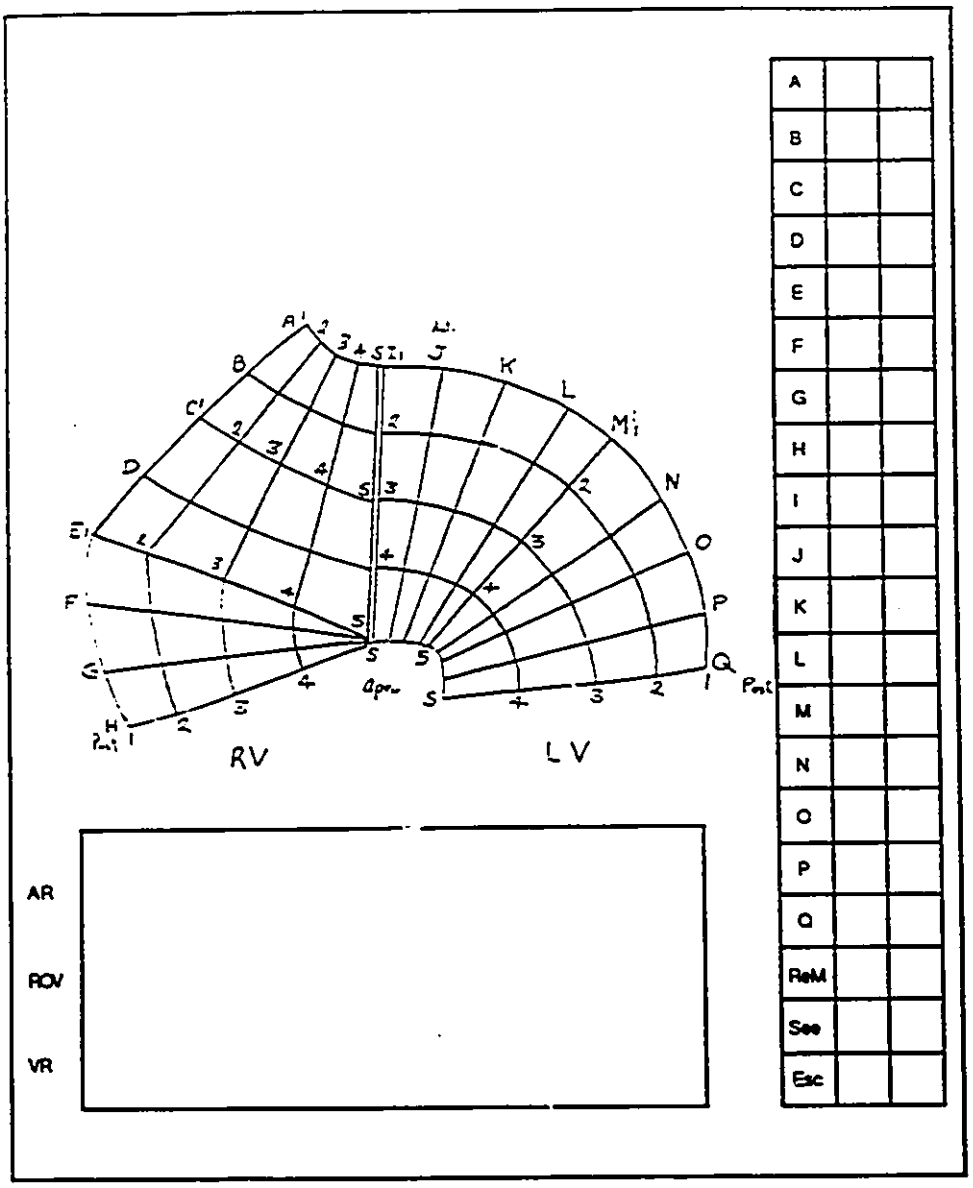

Figure E.2 Initial screen.

*** Data acquisition:**

To select a recording site, we just simply press one of the key from A to Q. The screen appears as shown in figure E.3.

* The difference between the activation time of the reference channel and that of the roving channel is 210 msec displayed on the second column of the table.

• The diference between the activation time of roving channel and the second reference channel is 4 msec displayed on the third column.

• The waveforms of the atrial reference, the roving and the ventricular reference channels are displayed in the window at the bottom of the screen.

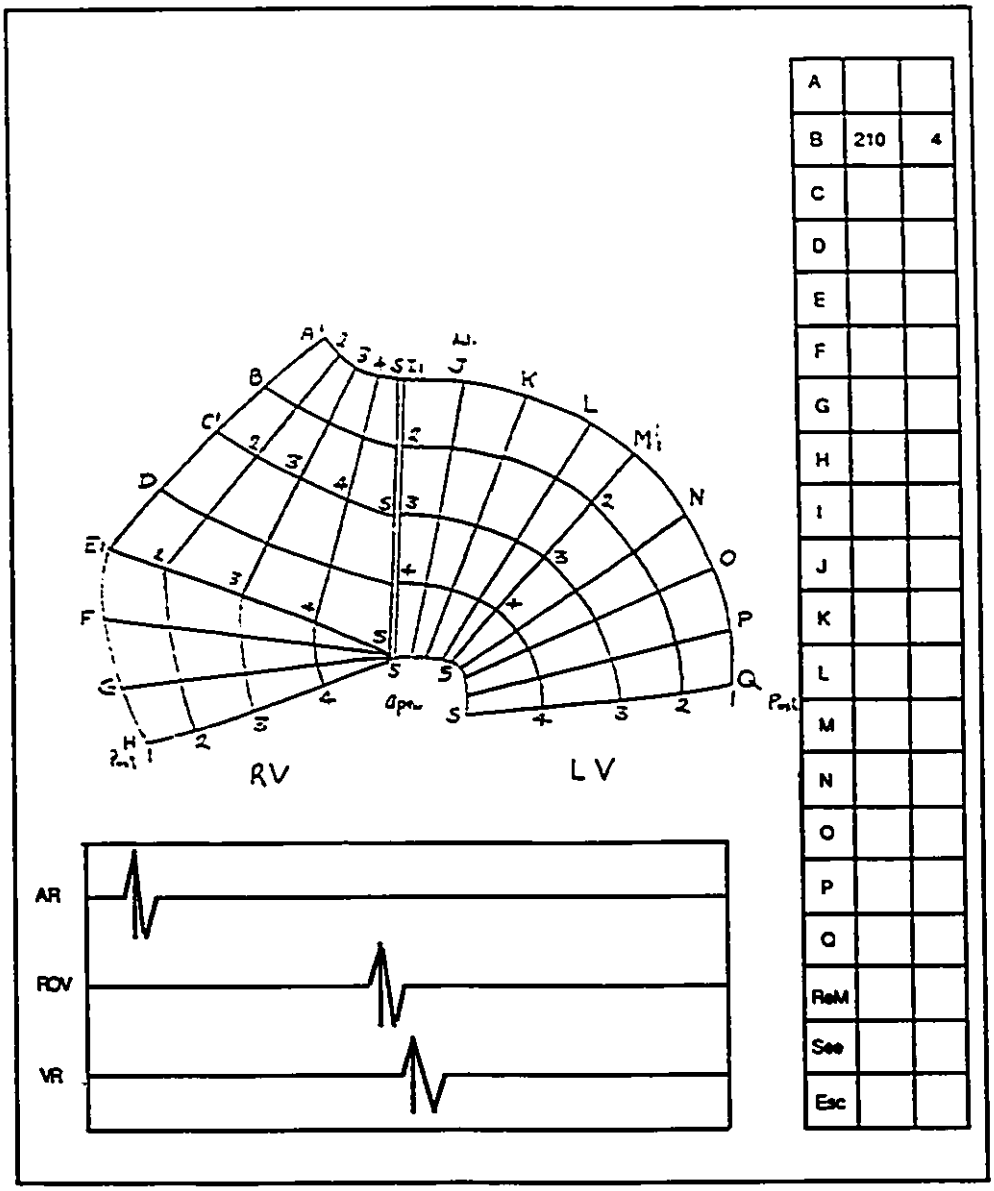

Figure E.3 Updating screen.

* Map of the activation time:

The map of the activation times can be displayed by typing S (stand for See) or we can move the pink cursor to the See location in the menu then press Return. The screen will be displayed as shown in Figure E.4.

* The smallest activation time at a particular location will be displayed in red color. For example in Figure E.4 , location L has the smallest activation time which is 140 ms.

* On the left hand side, the linear scale of the activation time versus the colors from red to blue is displayed.

* Each location on the ventricles is displayed with the corresponding color and from the map as shown in Figure E.4, the surgeons can see right away which location is late or early.

* At the same time, the activation time at all locations are stored in a file name "c:res.dat" for later examination.

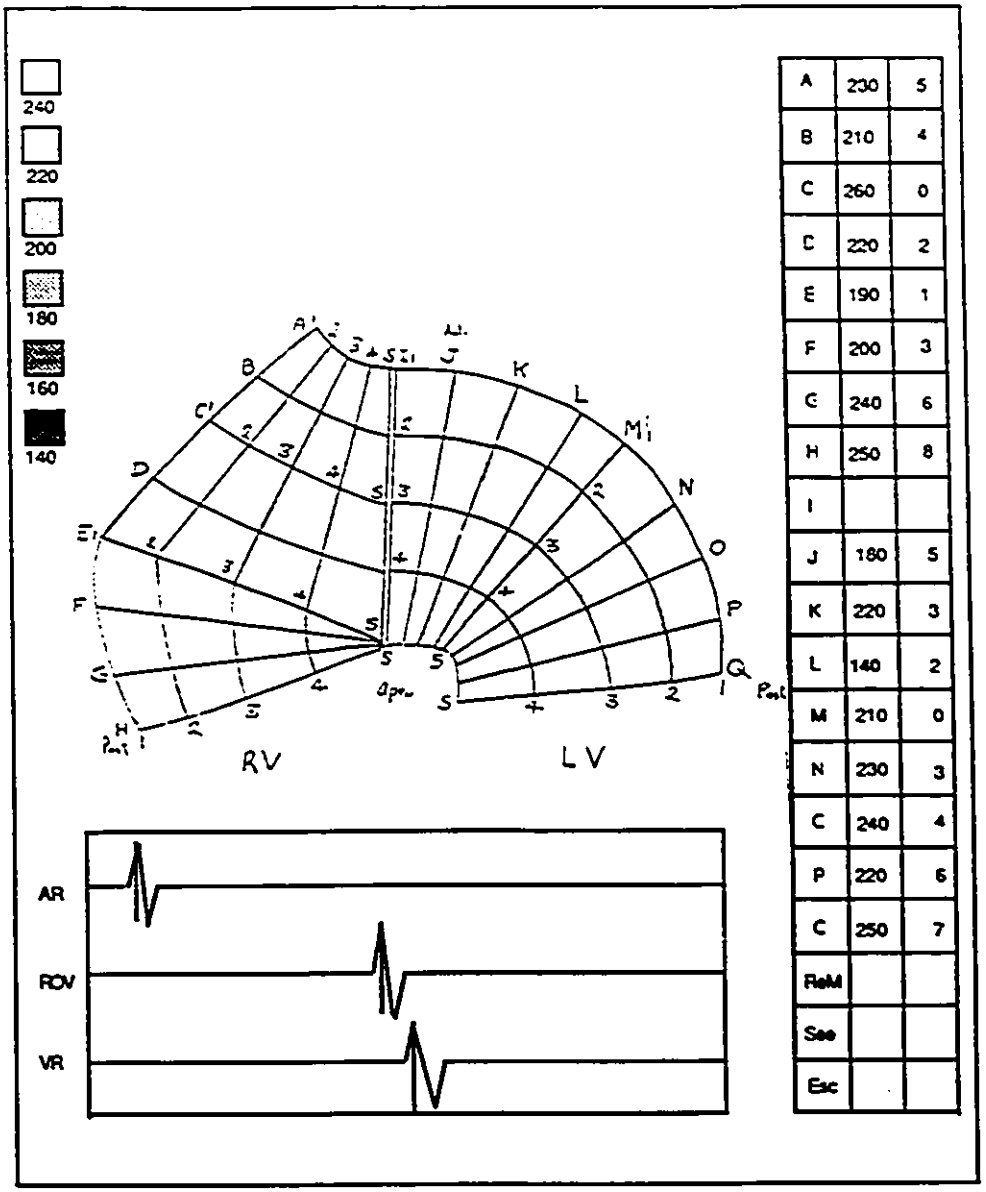

Figure E.4 Mapping screen.

*** Data storage:**

In order to store the waveform at a specific location for later study, we can press Enter if the pink cursor is already at the desired location. Otherwise, we can use the arrow key to move the yellow cursor to the interested location then press Enter. The segment of continuous 1.2 second of data at this location will be stored in the file "c:test.dat".

- **Data-retrieval:**

The program "dspl" are used to display the signals that were recorded during the operation:

The signals from the atrial reference, the roving and the ventricular reference channels are displayed as shown in Figure E.5.

At the lower left corner of the screen there are two numbers:

* The upper number represents the number of the set.

* The lower number represents the number of the left most sample in this set.

For an example as shown in Figure E.5, the number of the set is 9 and the left most sample is 500.

Typing F is to display the next set of data in the file.

Typing B is to display the previous set of data in the file.

Typing N is to display the next cycle of data in the set (there is usually two completed cycle in one segment).

Typing P is to display the previous cycle of data in the set.

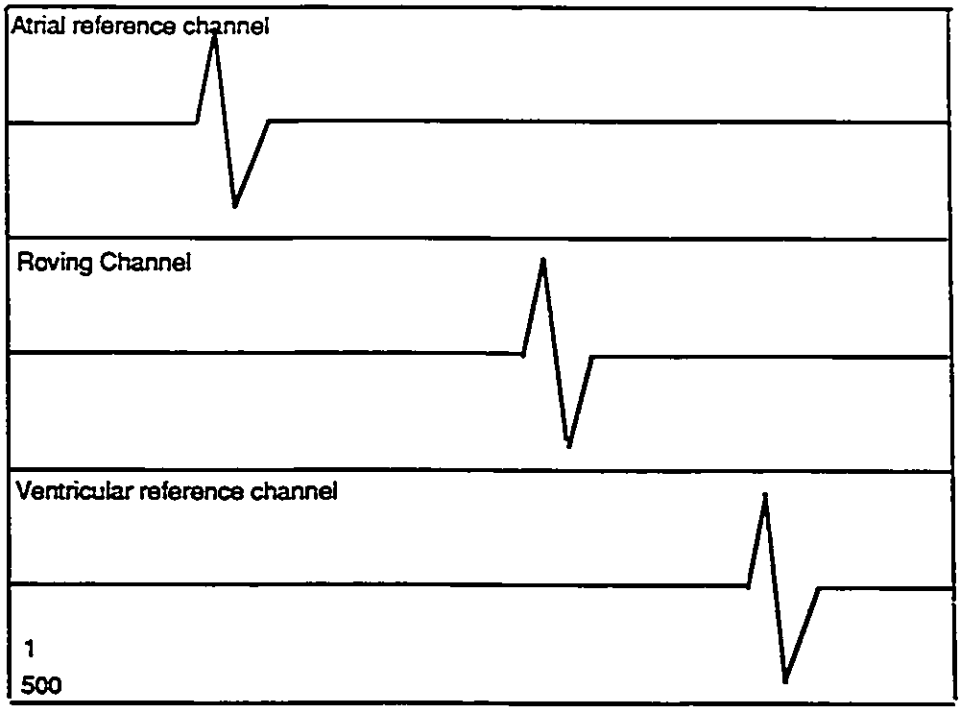

Figure E.5 Retrieval of the data from the file.

*** Correction of computer-estimated activation time:**

The waveform at each channel is displayed with indicator where the activation is detected. The user can modify the computer-estimated activation time by pressing R (stands for Remark) or using the arrow key to move the cursor to the ReM location in the menu then press Return. The pink cursor will move to the indicator of the atrial reference channel. Now the left and right arrow key can be used to move the pink cursor to the position which can be better to represent the activation time.

The up and down arrow key can be used to the move the pink cursor to the other channels to modify the activation times at the other channels.

In order to exit from this modification routine, we just simply press R again. The cursor now is moved back to the location in the menu where its waveform has just been modified.

BIBLIOGRAPHY

[1] H. J. J Wellens, F. W. M. M Bar, E. J. D. M Vanagt, and P. Brugada, " Medical treatment of ventricular tachycardiac: Considerations in the selection of patients for surgical treatment," *Amer. J. Cardiol.*, vol. 49, pp. 186-193, Jan. 1980.

[2] J. M. T. Debkker, M.J. Janse, J. F. L Van Capelle, and D.Durrer, "An interactive computer system for guiding the surgical treatment of life-threatening ventricular tachycardias," IEEE Trans. Biomed. Eng., vol. BME-31, pp. 362-368, 1984.

[3] G. Arisi, E. Macchi, S. Baruffi, S. Spaggiari, and B. Taccardi, "Potential fields on the ventricular surface of the exposed dog heart during normal excitation," *Circ. Res.*, vol. 52, pp 706-715,1983.

[4] J. P. Boineau, C. B. Miller, R. B. Schuessler, W. R. Roeske, L. J. Autry, A. C. Wylds, and D. A. Hill, "Activation sequence and potential distribution maps demonstrating multicentric atrial impulse origin in dogs," *Circ. Res.*, vol. 54, pp. 332-347, 1984.

[5] B. Taccardi, G. Arisi, E. Macchi, S. Baruffi, and S. Spaggiari, "A new intracavitary probe for detecting the site of origin of ectopic ventricular beats during one cardiac cycle," *Circulation*, vol. 75, pp. 272-281, 1987.

[6] A.L. Wit, M.A. Allessi, F.I.M. Bonke, W. Lammers, J. Smeets, and J.J. Fenoglio, Jr., "Electrophysiologic mapping to determine the mechanism of experimental ventricular tachycardia initiated by premature impulses: Experimental approach and initial results demonstrating reentrant excitation," *Amer. J. Cardiol.*, vol. 49, pp. 166-185, 1982.

[7] R.E. Ideker, W.M. Smith, A.G. Wallace, J. Kasell, L.A. Harrison, G.J. Klein, R.E. Kinicki, and J.J. Gallagher, " A computerized method for the rapid display of ventricular activation during the intraoperative study of arrhymias," *Circulation*, vol. 59, pp. 449-458, 1979.

[8] F.X. Witkowski, and P.B. Corr, "An automated simultaneous transmural cardiac mapping system," *American Physiological Society*, pp. H661-H667, 1984.

[9] I.D. Parson, and E. Downar, "Cardiac Instrumentation for the instantaneous display of endocardial and epicardial activation," *IEEE Trans. Biomed. Eng.*, Vol BME-34 No. 6, pp.468-472, 1987.

[10] J. J. Gallagher, M. Gilbert, R.H. Svenson, H. Miller, W.C. Sealy, J. Kasel, and A. G. Wallace, "Wolff-Pakinson-White syndrome: the problem, evaluation, and surgical correction," *Circulation* , vol. 51, pp.767-769, 1975.

[11] J.T. Bakker, M.J. Janse, F.V. Capelle, and D. Dukrrer, "Endocardial mapping by simultaneous recording of endocardial electrograms during cardiac surgery for ventricular aneurysm," *JACC*, Vol 2, No. 5, pp. 947-953, 1983.

[12] J.J. Gallagher, J. Kasell, W.C. Sealy, E.L.C. Pritchett, and A.G. Wallace, "Epicardial mapping in the Wolff-Parkinson-White syndrome," *Circulation*, Vol. 57, No. 5, pp. 855-865, May 1978.

[13] W.M. Smith, R.E. Idecker, R.E. Kinicki, and L. Harrison, "A computer system for the intraoperative mapping of ventricular arrhythmias," *Computers and Biomedical Research*, Vol. 13, pp.61-72, 1980.

[14] H.J. Vidaillet Jr., J.E. Lowe, L.D. German, P. Chen, G.S. Greer, M.R. Gilbert, W.M. Smith, S.J. Worley, and R.E. Ideker, "Computer-assisted intraoperative mapping system of the entire ventricular epicardium in the Wolff-Parkinson-White Syndrome," *American Journal of Cardialogy*, Vol 58, pp. 940-947, Nov.1, 1986.

[15] L. Harrison, R.E. Ideker, W.M. Smith, G.J. Klein, J. Kasell, A.G. Wallace, and J.J. Gallagher, "The sock electrode array: A tool for determining global epicardial activation during unstable arrythmias," *Pace*, vol. 3, pp. 531-540, 1980.

[16] J.B. Kramer, P.B.Corr, J.L. Cox, F.X. Witkowski, and M.E. Cain, "Simultaneous computer mapping to facilitate intraoperative localization of accessory pathways in patients with Wolff-Parkinson-White syndrome," *Amer. J. Cardiol.*, vol. 56, pp. 571-576, 1985.

[17] Canadian Standard Association, "Safety functions incorporating electronic technology," *Canadian Standard Association*, C222 No. 0.8-M1986, 1986.

[18] R.E. Ideker, W.M. Smith, P. Wolf, N.D. Danieley, and F.R. Bartram, " Simultaneous multichannel cardiac mapping systems," *Pace*, vol. 10, pp. 281-291, 1987.

[19] M.E. Josephson, L.N. Horowitz, A. Farshidi, J.F. Spear, J.A. Kastor, and E.N. Moore, "Recurrent sustained ventricular tachycardia," *Circulation*, Vol 57, No. 3, pp. 440-447, March 1978.

[20] S.M. Blanchard, W.M. Smith, R.J. Damiano, D.W. Molter, R.E. Ideker, and J.E. Lowe, "Four digital algorithms for activation detection from unipolar epicardial electrograms," *IEEE Trans. Biomed. Eng.*, vol. BME-36, pp. 256-261, 1989.

[21] J.J. Gallagher, J.H. Kasell, J.L. Cox, W.M. Smith, R.E. Ideker, and W.M. Smith, "Techniques of intraoperative electraphysiologic mapping," *American Journal of Cardiology*, Vol. 49, pp. 221-235, Jan 1982.

[22] A.E. Marble, C.M. McIntyre, R. Hasting-James, and C.W. Hor, "A comparison of digital algorithms used in computing the derivative of left ventricular pressure," *IEEE Trans. Biomed. Eng.*, vol. BME-28, pp. 524-528, 1981.

[23] R.E. Ideker, W.M. Smith, S.M. Blanchard, S.L. Reiser, E.V. Simpson, P.D. Wolf, and N.D. Danieley, "The assumptions of isochronal cardiac mapping," *Pace*, Vol 12, pp. 456-477, 1989.

[24] A. C. Guyton, "Human Physiology and mechanism of disease", *W. B. Saunders*, 4th edition, 1987.

[25] R.C. Barr, and M.S. Spach, "Sampling rates required for digital recording of intracellular and extracellular cardiac potentials," *Circulation*, Vol. 55, No. 1, pp. 40-48, 1977.

[26] E. J. Kennedy, "Operational amplifier circuits theory and applications", *Holt, Rinehart and Winston, Inc.*, 1988.

[27] F.J. Claydon, T.C. Pilkington, and R.E. Ideker, "Classification of heart tissue from bipolar and unipolar intramural potentials," *IEEE Trans Biomed Eng*, Vol BME-32, No.7, pp.513-520, 1985.

[28] AD1000 User's Manual, *Real time Devices*, 1989.

[29] H. Itatsu, "Theoretical interpolation of contiguous bipolar ECG and its relationship to the time of arrival of activation," *Jpn Circ J*, vol.18, pp. 1-10, 1954.

[30] E.V. Simpson, R.E. Ideker, and W.M. Smith, " An automatic activation detector for bipolar cardiac electrograms," *Proc 10th Annual Conf of the IEEE Engineering in medicine and biology society*, pp. 113-144, 1988.

[31] S.M. Blanchard, R.J. Damiano, T. Asano, W.M. Smith, R.E. Ideker, and J.E. Lowe , "The effects of distant cardiac events on local activation in unipoar epicardial electrograms," *IEEE Trans Biomed Eng*, Vol. BME-34, No. 7, pp. 539-546, 1987.

[32] A. Haber and R.P. Runyon, "General Statistics", *Addison-Wesley Publishing Company*, 2nd Eddition, 1973.

[33] F. Mosteller, R.E.K. Rourke, and G.B. Thomas, "Probability With Statistical Applications", *Addison-Wesley Publishing Company*, 2nd Eddition, 1965.

[34] The Open University, "Significant testing applied non-parametric Statistics Quality Control", *The Open University Press*, 1st Eddition, 1974.

[35] I. Miller and J.E. Freund, "Probability and statistics for engineers", *Prentice-hall, Inc.*, 3rd Eddition, 1985.