1

Introduction to Computers in Medicine

Willis J. Tompkins

The field of computers in medicine is quite broad. We can only cover a small part of it in this book. We choose to emphasize the importance of real-time signal processing in medical instrumentation. This chapter discusses the nature of medical data, the general characteristics of a medical instrument, and the field of medicine itself. We then go on to review the history of the microprocessor-based system because of the importance of the microprocessor in the design of modern medical instruments. We then give some examples of medical instruments in which the microprocessor has played a key role and in some cases has even empowered us to develop new instruments that were not possible before. The chapter ends with a discussion of software design and the role of the personal computer in development of medical instruments.

1.1 CHARACTERISTICS OF MEDICAL DATA

Figure 1.1 shows the three basic types of data that must be acquired, manipulated, and archived in the hospital. Alphanumeric data include the patient's name and address, identification number, results of lab tests, and physicians' notes. Images include Xrays and scans from computer tomography, magnetic resonance imaging, and ultrasound. Examples of physiological signals are the electrocardiogram (ECG), the electroencephalogram (EEG), and blood pressure tracings.

Quite different systems are necessary to manipulate each of these three types of data. Alphanumeric data are generally managed and organized into a database using a general-purpose mainframe computer.

Image data are traditionally archived on film. However, we are evolving toward picture archiving and communication systems (PACS) that will store images in digitized form on optical disks and distribute them on demand over a high-speed local area network (LAN) to very high resolution graphics display monitors located throughout a hospital.

On the other hand, physiological signals like those that are monitored durin surgery in the operating room require real-time processing. The clinician mus know immediately if the instrument finds abnormal readings as it analyzes th continuous data.

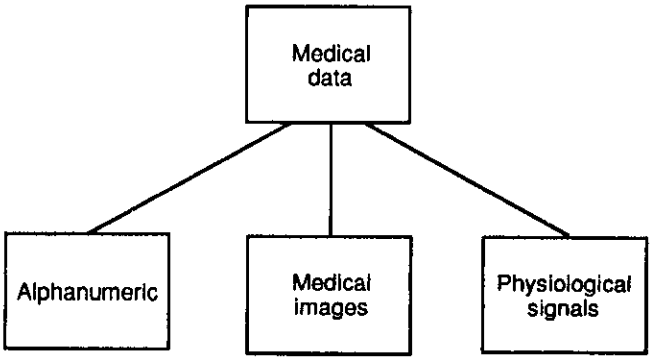


Figure 1.1 Types of medical data.

It is this final type of data on which we concentrate in this book. One of the mos monitored signals is the ECG, so we use it as the example signal to process i many examples.

1.2 WHAT IS A MEDICAL INSTRUMENT?

There are many different types of medical instruments. The ones on which we con centrate in this book are those that monitor and analyze physiological signals fron a patient. Figure 1.2 shows a block diagram that characterizes such instruments Sensors measure the patient's physiological signals and produce electrical signal (generally time-varying voltages) that are analogs of the actual signals.

A set of electrodes may be used to sense a potential difference on the body sur face such as an ECG or EEG. Sensors of different types are available to transduc into voltages such variables as body core temperature and arterial blood pressure The electrical signals produced by the sensors interface to a processor which is re sponsible for processing and analysis of the signals. The processor block typicall includes a microprocessor for performing the necessary tasks. Many instrument have the ability to display, record, or distribute through a network either the raw signal captured by the processor or the results of its analysis. In some instruments the processor performs a control function. Based on the results of signal analysis the processor might instruct a controller to do direct therapeutic intervention on a

patient (closed loop control) or it may signal a person that there is a problem that requires possible human intervention (open loop control).

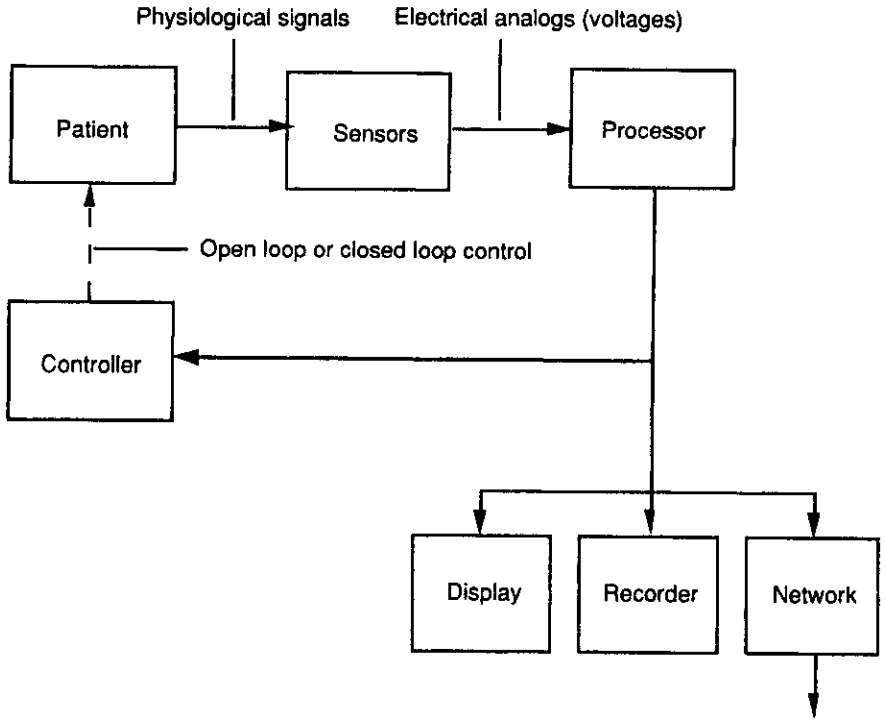


Figure 1.2 Basic elements of a medical instrumentation system.

Let us consider two types of medical instrumentation and see how they fit this block diagram. The first is an intensive care unit (ICU) system, a large set of instrumentation that monitors a number of patients simultaneously. The second is a cardiac pacemaker so small that it must fit inside the patient.

In the case of the ICU, there are normally several sensors connected to each patient receiving intensive care, and the processor (actually usually more than one processor) monitors and analyzes all of them. If the processor discovers an abnormality, it alerts the medical staff, usually with audible alarms. A display permits the staff to see raw data such as the ECG signals for each patient and also data obtained from the analysis such as numerical readouts of heart rate and blood pressure. The network connects the bedside portion of the instrumentation to a central console in the ICU. Another network might connect the ICU system to other databases remotely located in the hospital. An example of a closed loop device that

is sometimes used is an infusion pump. Sensors monitor fluid loss as the amount of urine collected from the patient, then the processor instructs the pump to infuse the proper amount of fluid into the patient to maintain fluid balance, thereby acting as a therapeutic device.

Now consider Figure 1.2 for the case of the implanted cardiac pacemaker. The sensors are electrodes mounted on a catheter that is placed inside the heart. The processor is usually a specialized integrated circuit designed specifically for this ultra-low-power application rather than a general-purpose microprocessor. The processor monitors the electrogram from the heart and analyzes it to determine if the heart is beating by itself. If it sees that the heart goes too long without its own stimulus signal, it fires an electrical stimulator (the controller in this case) to inject a large enough current through the same electrodes as those used for monitoring. This stimulus causes the heart to beat. Thus this device operates as a closed loop therapy delivery system. The early pacemakers operated in an open loop fashion, simply driving the heart at some fixed rate regardless of whether or not it was able to beat in a normal physiological pattern most of the time. These devices were far less satisfactory than their modern *intelligent* cousins. Normally a microprocessor-based device outside the body placed over a pacemaker can communicate with it through telemetry and then display and record its operating parameters. Such a device can also set new operating parameters such as amplitude of current stimulus. There are even versions of such devices that can communicate with a central clinic over the telephone network.

Thus, we see that the block diagram of a medical instrumentation system serves to characterize many medical care devices or systems.

1.3 ITERATIVE DEFINITION OF MEDICINE

Figure 1.3 is a block diagram that illustrates the operation of the medical care system. Data collection is the starting point in health care. The clinician asks the patient questions about medical history, records the ECG, and does blood tests and other tests in order to define the patient's problem. Of course medical instruments help in some aspects of this data collection process and even do some preprocessing of the data. Ultimately, the clinician analyzes the data collected and decides what is the basis of the patient's problem. This decision or diagnosis leads the clinician to prescribe a therapy. Once the therapy is administered to the patient, the process continues around the closed loop in the figure with more data collection and analysis until the patient's problem is gone.

The function of the medical instrument of Figure 1.2 thus appears to be a model of the medical care system itself.

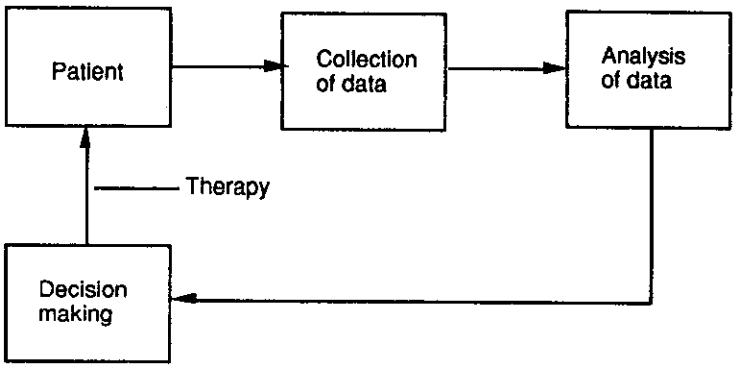


Figure 1.3 Basic elements of a medical care system.

1.4 EVOLUTION OF MICROPROCESSOR-BASED SYSTEMS

In the last decade, the microcomputer has made a significant impact on the design of biomedical instrumentation. The natural evolution of the microcomputer-based instrument is toward more *intelligent* devices. More and more computing power and memory are being squeezed into smaller and smaller spaces. The commercialization of laptop PCs with significant computing power has accelerated the technology of the battery-powered, patient-worn portable instrument. Such an instrument can be truly a *personal* computer looking for problems specific to a given patient during the patient's daily routines. The ubiquitous PC itself evolved from minicomputers that were developed for the biomedical instrumentation laboratory, and the PC has become a powerful tool in biomedical computing applications. As we look to the future, we see the possibility of developing instruments to address problems that could not be previously approached because of considerations of size, cost, or power consumption.

The evolution of the microcomputer-based medical instrument has followed the evolution of the microprocessor itself (Tompkins and Webster, 1981). Figure 1.4 shows a plot of the number of transistors in Intel microprocessors as a function of time. The microprocessor is now more than 20 years old. It has evolved from modest beginnings as an integrated circuit with 2,000 transistors (Intel 4004) in 1971 to the powerful central processing units of today having more than 1,000,000 transistors (e.g., Intel i486 and Motorola 68040). One of the founders of Intel named Moore observed that the number of functional transistors that can be put on a single piece of silicon doubles about every two years. The solid line in the figure represents this observation, which is now known as Moore's Law. The figure shows that Intel's introduction of microprocessors, to date, has followed Moore's Law exceptionally well. The company has predicted that they will be able to continue

producing microprocessors with this exponential growth in the number of transistors per microprocessor until at least the end of this century. Thus, in less than a decade, the microprocessor promises to become superpowerful as a parallel processing device with 100 million transistors on one piece of silicon. It most likely will be more powerful than any of today's supercomputers, will certainly be part of a desktop computer, and possibly will be powerable by batteries so that it can be used in portable devices.

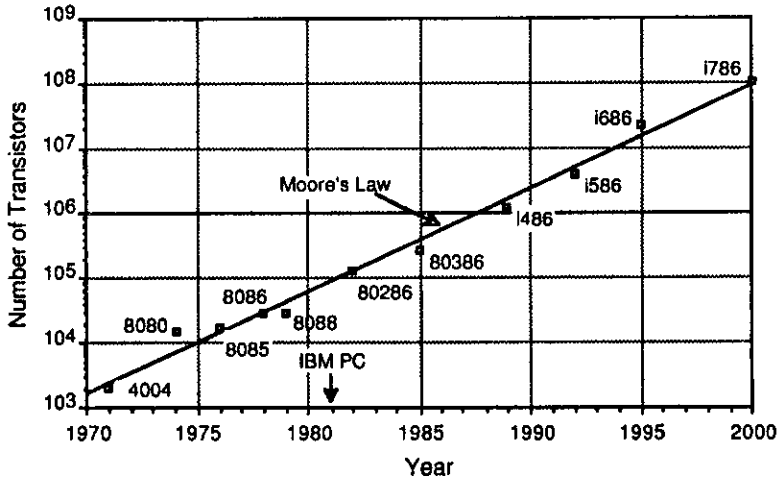


Figure 1.4 The evolution of the microprocessor. The number of transistors in a microprocessor has increased exponentially throughout the history of the device. The trend is expected to continue into the future.

The evolution of the microprocessor from its early beginnings in 1971 as a primitive central processing unit to the powerful component of today has made a significant impact on the design of biomedical instrumentation. More computing power and memory are being squeezed into fewer integrated circuits to provide increasingly more powerful instruments. The PC itself has become a powerful tool in biomedical computing applications. In the future, we will be able to develop new medical instruments to address problems that were previously not solvable. This possibility exists because microprocessor-based systems continuously increase in computing power and memory while decreasing in size, cost, and power consumption.

1.4.1 Evolution of the personal computer

Figure 1.5 shows the history of the development of the computer from the first mechanical computers such as those built by Charles Babbidge in the 1800s to the modern personal computers, the IBM PC and the Apple Macintosh. The only computers prior to the twentieth century were mechanical, based on gears and mechanical linkages.

In 1941 a researcher named Atanasoff demonstrated the first example of an electronic digital computer. This device was primitive even compared to today's four-function pocket calculator. The first serious digital computer called ENIAC (Electronic Numerical Integrator And Calculator) was developed in 1946 at the Moore School of Electrical Engineering of the University of Pennsylvania. Still simple compared to the modern PC, this device occupied most of the basement of the Moore School and required a substantial air conditioning system to cool the thousands of vacuum tubes in its electronic brain.

The invention of the transistor led to the Univac I, the first commercial computer. Several other companies including IBM subsequently put transistorized computers into the marketplace. In 1961, researchers working at Massachusetts Institute of Technology and Lincoln Labs used the technology of the time to build a novel minicomputer quite unlike the commercial machines. This discrete-component, transistorized minicomputer with magnetic core memory called the LINC (Laboratory INstrument Computer) was the most significant historical development in the evolution of the PC.

The basic design goal was to transform a general-purpose computer into a laboratory instrument for biomedical computing applications. Such a computer, as its designers envisioned, would have tremendous versatility because its function as an instrument could be completely revised simply by changing the program stored in its memory. Thus this computer would perform not only in the classical computer sense as an equation solving device, but also by reprogramming (software), it would be able to mimic many other laboratory instruments.

The LINC was the most successful minicomputer used for biomedical applications. In addition, its design included features that we have come to expect in modern PCs. In particular, it was the world's first interactive computer. Instead of using punched cards like the other computers of the day, the LINC had a keyboard and a display so that the user could sit down and program it directly. This was the first digital computer that had an interactive graphics display and that incorporated knobs that were functionally equivalent to the modern joystick. It also had built-in signal conversion and instrument interfacing hardware, with a compact, reliable digital tape recorder, and with sound generation capability. You could capture an ECG directly from a patient and show the waveform on the graphics display.

The LINC would have been the first personal computer if it had been smaller (it was about the size of a large refrigerator) and less expensive (it cost about $50,000 in kit form). It was the first game computer. Programmers wrote software for a two-player game called Spacewar. Each player controlled the velocity and

direction of a spaceship by turning two knobs. Raising a switch fired a missile at the opposing ship. There were many other games such as pong and music that included an organ part from Bach as well as popular tunes.

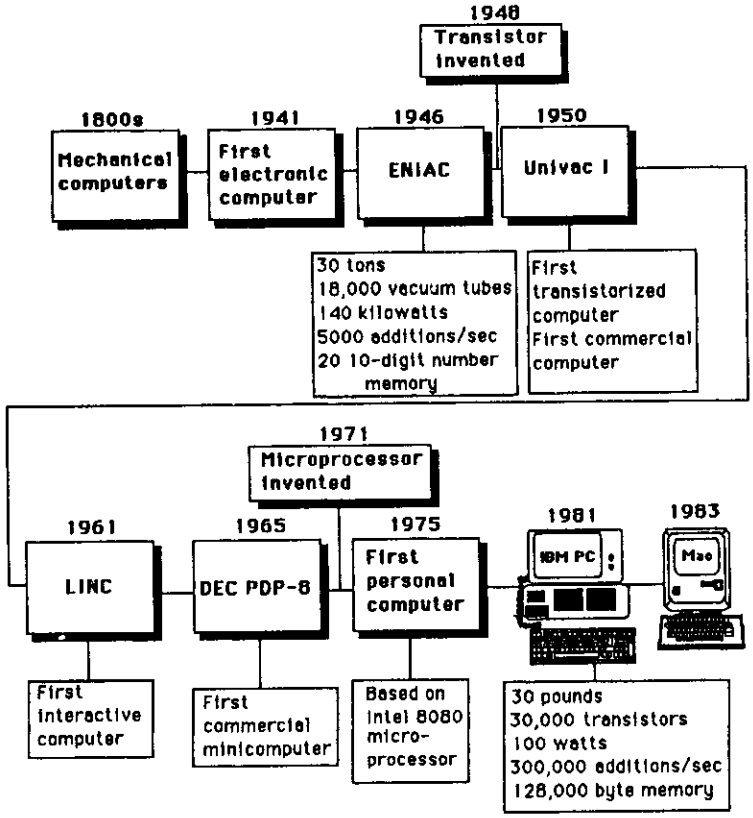


Figure 1.5 The evolution of the personal computer.

The LINC was followed by the world's first commercial minicomputer, which was also made of discrete components, the Digital Equipment Corporation PDP-8. Subsequently Digital made a commercial version of the LINC by combining the LINC architecture with the PDP-8 to make a LINC-8. Digital later introduced a more modern version of the LINC-8 called the PDP-12. These computers were phased out of Digital's product line some time in the early 1970s. I have a special fondness for the LINC machines since a LINC-8 was the first computer that I programmed that was interactive, could display graphics, and did not require the use of awkward media like punched cards or punched paper tape to program it. One of

the programs that I wrote on the LINC-8 in the late 1960s computed and displayed the vectorcardiogram loops of patients (see Chapter 2). Such a program is easy to implement today on the modern PC using a high-level computing language such as Pascal or C.

Although invented in 1971, the first microprocessors were poor central processing units and were relatively expensive. It was not until the mid-1970s when useful 8-bit microprocessors such as the Intel 8080 were readily available. The first advertised microcomputer for the home appeared on the cover of *Popular Electronics Magazine* in January 1975. Called the Altair 8800, it was based on the Intel 8080 microprocessor and could be purchased as a kit. The end of the decade was full of experimentation and new product development leading to the introduction of PCs like the Apple II and microcomputers from many other companies.

1.4.2 The ubiquitous PC

A significant historical landmark was the introduction of the IBM PC in 1981. On the strength of its name alone, IBM standardized the personal desktop computer. Prior to the IBM PC, the most popular computers used the 8-bit Zilog Z80 microprocessor (an enhancement of the Intel 8080) with an operating system called CP/M (Control Program for Microprocessors). There was no standard way to format a floppy disk, so it was difficult to transfer data from one company's PC to another. IBM singlehandedly standardized the world almost overnight on the 16-bit Intel 8088 microprocessor, Microsoft DOS (Disk Operating System), and a uniform floppy disk format that could be used to carry data from machine to machine. They also stimulated worldwide production of IBM PC compatibles by many international companies. This provided a standard computing platform for which software developers could write programs. Since so many similar computers were built, inexpensive, powerful application programs became plentiful. This contrasted to the minicomputer marketplace where there are relatively few similar computers in the field, so a typical program is very expensive and the evolution of the software is relatively slow.

Figure 1.6 shows how the evolution of the microprocessor has improved the performance of desktop computers. The figure is based on the idea that a complete PC at any given time can be purchased for about $5,000. For this cost, the number of MIPS (million instructions per second) increases with every new model because of the increasing power of the microprocessor. The first IBM PC introduced in 1981 used the Intel 8088 microprocessor, which provided 0.1 MIPS. Thus, it required 10 PCs at $5,000 each or $50,000 to provide one MIPS of computational power. With the introduction of the IBM PC/AT based on the Intel 80286 microprocessor, a single $5,000 desktop computer provided one MIPS. The most recent IBM PS computers using the Intel i486 microprocessor deliver 10 MIPS for that same $5,000. A basic observation is that the cost per MIPS of computing power is decreasing logarithmically in desktop computers.

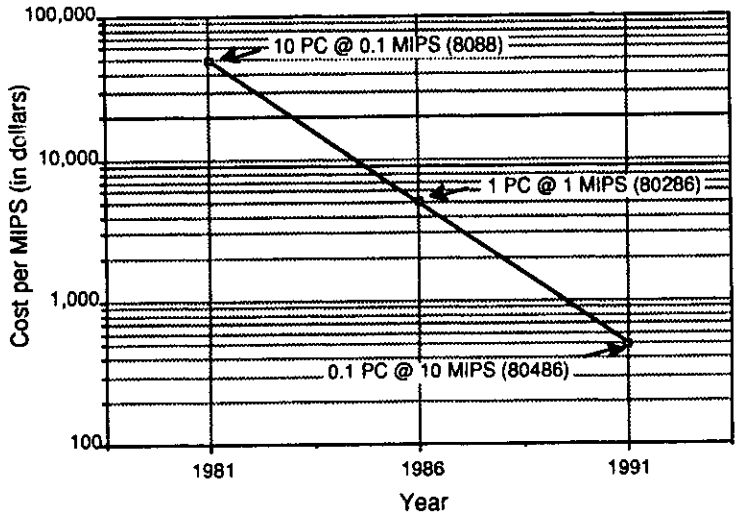


Figure 1.6 The inverse relationship between computing power and cost for desktop computers. The cost per MIPS (million instructions per second) has decreased logarithmically since the introduction of the IBM PC.

Another important PC landmark was the introduction of the Apple Macintosh in 1983. This computer popularized a simple, intuitive user-to-machine interface. Since that time, there have been a number of attempts to implement similar types of graphical user interface (GUI) for the IBM PC platform, and only recently have practical solutions come close to reality.

More than a decade has elapsed since the introduction of the IBM PC, and most of the changes in the industry have been evolutionary. Desktop PCs have continued to evolve and improve with the evolution of the technology, particularly the microprocessor itself. We now have laptop and even palmtop PC compatibles that are portable and battery powered. We can carry around a significant amount of computing power wherever we go.

Figure 1.7 shows how the number of electronic components in a fully functional PC has decreased logarithmically with time and will continue to decrease in the future. In 1983, about 300 integrated circuits were required in each PC, of which about half were the microprocessor and support logic and the other half made up the 256-kbyte memory. Today half of the parts are still dedicated to each function, but a complete PC can be built with about 18 ICs. In the past six years, the chip count in a PC has gone from about 300 integrated circuits in a PC with a 256-kbyte memory to 18 ICs in a modern 2-Mbyte PC. By the mid-1990s, it is likely that a PC with 4-Mbyte memory will be built from three electronic components, a single IC for all the central processing, a read-only memory (ROM) for the basic

input/output software (BIOS), and a 4-Mbyte dynamic random access memory (DRAM) chip for the user's program and data storage. Thus the continuing trend is toward more powerful PCs with more memory in smaller space for lower cost and lower power consumption.

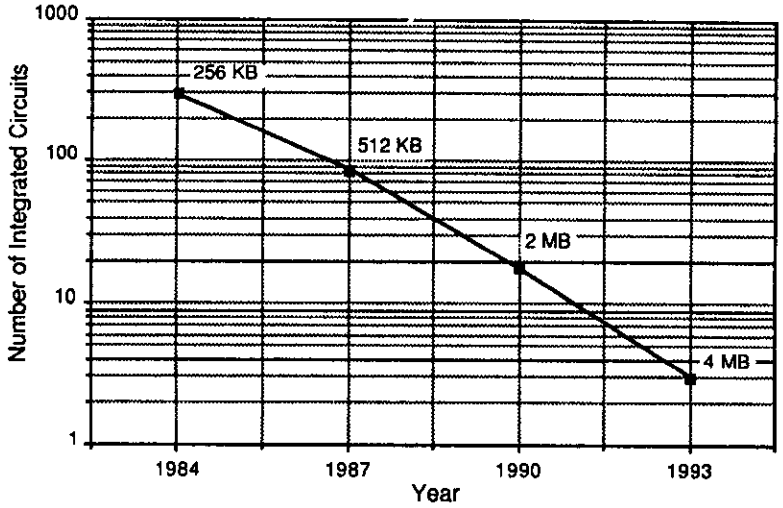


Figure 1.7 The number of components in a PC continues to decrease while the computing performance increases.

In the 1970s, the principal microprocessors used in desktop computers as well as other systems including medical instruments were 8-bit microprocessors. The 1980s were dominated by the 16-bit microprocessors. The 1990s were launched with the 32-bit processor, but the technology's exponential growth will likely lead to useful new architectures on single ICs, such as parallel processors and artificial neural networks.

Figure 1.8 compares the modern PC with the human brain. The figure provides information that gives us insight into the relative strengths and weaknesses of the computer compared to the brain. From the ratios provided, we can clearly see that the personal computer is one or more orders of magnitude heavier, larger, and more power consuming than the human brain. The PC has several orders of magnitude fewer functional computing elements and memory cells in its "brain" than does the human brain.

COMPARISON OF PERFORMANCE OF IBM PC AND HUMAN BRAIN

System	Weight (lbs)	Size (ft^3)	Power (watts)	CPU elements	Memory (bits)	Conduction rate (impulses/s)	Benchmark (additions/s)
IBM PC	30	5	200	10^6 transistors (or equiv.)	10^7	10^5	10^6
Brain	3	0.05	10	10^{10} neurons	10^{20}	10^2	1
Ratio (PC/brain)	10	100	20	10^{-4}	10^{-13}	10^3	10^6

Figure 1.8 The PC and the human brain each have their own characteristic strengths and weaknesses.

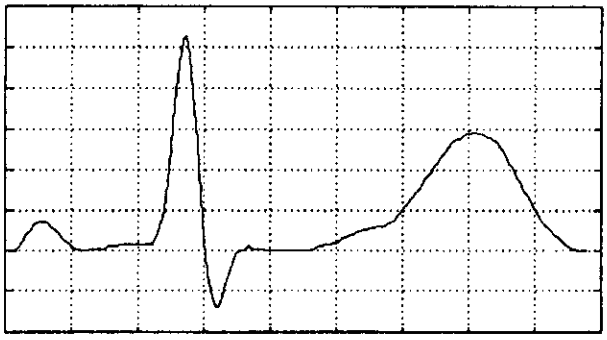
Then what good is it? The answer lies in the last two columns of the figure. The speed at which impulses are conducted in the computer so far exceeds the speed of movement of information within the brain that the PC has a very large computation speed advantage over the brain. This is illustrated by a benchmark which asks a human to add one plus one plus one and so on, reporting the sum after each addition. The PC can do this computational task about one million times faster than the human. If the PC is applied to tasks that exploit this advantage, it can significantly outperform the human.

Figure 1.9(a) is an array of numbers, the basic format in which all data must be placed before it can be manipulated by a computer. These numbers are equally spaced amplitude values for the electrocardiogram (ECG) of Figure 1.9(b). They were obtained by sampling the ECG at a rate of 200 samples per second with an analog-to-digital converter. This numerical representation of the ECG is called a digital signal. The human eye-brain system, after years of experience in learning the characteristics of these signals, is particularly good at analyzing the analog waveform itself and deducing whether such a signal is normal or abnormal.

On the other hand, the computer must analyze the array of numbers using software algorithms in order to make deductions about the signal. These algorithms typically include digital signal processing, which is the emphasis of this book, together with decision logic in order to analyze biomedical signals as well as medical images. It is the enormous speed of the computer at manipulating numbers that makes many such algorithms possible.

0 0 0 0 2 5 8 10 13 14 14 14 12 11 9 7 5 4 2 1 1 0 0 1 1 1 1 1 2 2 2 3 3 3 3 3 3 3 3
3 3 6 11 20 33 51 72 91 103 105 96 77 53 27 5 -11 -23 -28 -28 -23 -17 -10 -5 -1
0 1 2 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 2 2 3 3 4 4 5 6 7 8 8 9 10 10 11 11 12 12
12 13 14 16 18 20 22 24 27 29 31 34 37 39 42 44 47 49 52 54 55 56 57 57 58 58
57 57 56 56 54 52 50 47 43 40 36 33 29 26 23 20 17 14 12 10 8 7 5 3 2 1 1 0 0 0

(a)



(b)

Figure 1.9 Two views of an electrocardiogram. (a) The computer view is an array of numbers that represent amplitude values as a function of time. (b) The human view is a time-varying waveform.

1.5 THE MICROCOMPUTER-BASED MEDICAL INSTRUMENT

The progress in desktop and portable computing in the past decade has provided the means with the PC or customized microcomputer-based instrumentation to develop solutions to biomedical problems that could not be approached before. One of our personal interests has been the design of portable instruments that are light, compact, and battery powered (Tompkins, 1981). A typical instrument of this type is truly a personal computer since it is programmed to monitor signals from transducers or electrodes mounted on the person who is carrying it around.

1.5.1 Portable microcomputer-based instruments

One example of a portable device is the portable arrhythmia monitor which monitors a patient's electrocardiogram from chest electrodes and analyzes it in real time to determine if there are any heart rhythm abnormalities. We designed a prototype of such a device more than a decade ago (Tompkins, 1978). Because of the technology available at that time, this device was primitive compared with modern commercially available portable arrhythmia monitors. The evolution of the technology also permits us to think of even more extensions that we can make. Instead

the hospital, we can now think of designing a device that would help diagnose the heart abnormality when the patient arrives in the emergency room. With a careful design, the same device might go with the patient to monitor the cardiac problem during surgery in the operating room, continuously learning the unique characteristics of that patient's heart rhythms. The device could follow the patient throughout the hospital stay, alerting the hospital staff to possible problems in the intensive care unit, in the regular hospital room, and even in the hallways as the patient walks to the cafeteria. The device could then accompany the patient home, providing continuous monitoring that is not now practical to do, during the critical times following open heart surgery (Tompkins, 1988). Chapter 13 discusses the concept of a portable arrhythmia monitor in greater detail.

There are many other examples of portable biomedical instruments in the marketplace and in the research lab. One other microcomputer-based device that we contributed to developing is a calculator-size product called the CALTRAC that uses a miniature accelerometer to monitor the motion of the body. It then converts this activity measurement to the equivalent number of calories and displays the cumulative result on an LCD display (Doumas et al., 1982). There is now an implanted pacemaker that uses an accelerometer to measure the level of a patient's activity in order to adjust the pacing rate.

We have also developed a portable device that monitors several pressure channels from transducers on a catheter placed in the esophagus. It analyzes the signals for pressure changes characteristic of swallowing, then records these signals in its semiconductor memory for later transfer to an IBM PC where the data are further analyzed (Pfister et al., 1989).

Another portable device that we designed monitors pressure sensors placed in the shoes to determine the dynamic changes in pressure distribution under the foot for patients such as diabetics who have insensate feet (Mehta et al., 1989).

1.5.2 PC-based medical instruments

The economy of mass production has led to the use of the desktop PC as the central computer for many types of biomedical applications. Many companies use PCs for such applications as sampling and analyzing physiological signals, maintaining equipment databases in the clinical engineering department of hospitals, and simulation and modeling of physiological systems.

You can configure the PC to have user-friendly, interactive characteristics much like the LINC. This is an important aspect of computing in the biomedical laboratory. The difference is that the PC is a much more powerful computer in a smaller, less expensive box. Compared to the LINC of two decades ago, the PC has more than 100 times the computing power and 100 times the memory capacity in one-tenth the space for one-tenth the cost. However, the LINC gave us tremendous insight into what the PC should be like long before it was possible to build a personal computer.

We use the PC as a general-purpose laboratory tool to facilitate research on many biomedical computing problems. We can program it to execute an infinite variety of programs and adapt it for many applications by using custom hardware interfaces. For example, the PC is useful in rehabilitation engineering. We have designed a system for a blind person that converts visible images to tactile (touch) images. The PC captures an image from a television camera and stores it in its memory. A program presents the image piece by piece to the blind person's fingertip by activating an array of tactors (i.e., devices that stimulate the sense of touch) that are pressed against his/her fingertip. In this way, we use the PC to study the ability of a blind person to "see" images with the sense of touch (Kaczmarek et al., 1985; Frisken-Gibson et al., 1987).

One of the applications that we developed based on an Apple Macintosh II computer is electrical impedance tomography—EIT (Yorkey et al., 1987; Woo et al., 1989; Hua et al., 1991). Instead of the destructive radiation used for the familiar computerized tomography techniques, we inject harmless high-frequency currents into the body through electrodes and measure the resistances to the flow of electricity at numerous electrode sites. This idea is based on the fact that body organs differ in the amount of resistance that they offer to electricity. This technology attempts to image the internal organs of the human body by measuring impedance through electrodes placed on the body surface.

The computer controls a custom-built 32-channel current generator that injects patterns of high-frequency (50-kHz) currents into the body. The computer then samples the body surface voltage distribution resulting from these currents through an analog-to-digital converter. Using a finite element resistivity model of the thorax and the boundary measurements, the computer then iteratively calculates the resistivity profile that best satisfies the measured data. Using the standard graphics display capability of the computer, an image is then generated of the transverse body section resistivity. Since the lungs are high resistance compared to the heart and other body tissues, the resistivity image provides a depiction of the organ system in the body. In this project the Macintosh does all the instrumentation tasks including control of the injected currents, measurement of the resistivities, solving the computing-intensive algorithms, and presenting the graphical display of the final image.

There are many possible uses of PCs in medical instrumentation (Tompkins, 1986). We have used the IBM PC to develop signal processing and artificial neural network (ANN) algorithms for analysis of the electrocardiogram (Pan and Tompkins, 1985; Hamilton and Tompkins, 1986; Xue et al., 1992). These studies have also included development of techniques for data compression to reduce the amount of storage space required to save ECGs (Hamilton and Tompkins, 1991a, 1991b).

1.6 SOFTWARE DESIGN OF DIGITAL FILTERS

In addition to choosing a personal computer hardware system for laboratory use, we must make additional software choices. The types of choices are frequently closely related and limited by the set of options available for a specific hardware system. Figure 1.10 shows that there are three levels of software between the hardware and the real-world environment: the operating system, the support software, and the application software (the shaded layer). It is the application software that makes the computer behave as a medical instrument. Choices of software at all levels significantly influence the kinds of applications that a system can address.

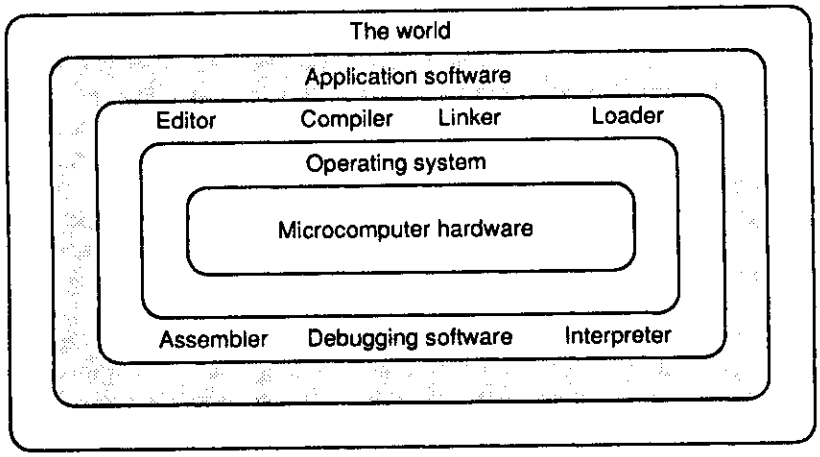


Figure 1.10 Three levels of software separate a hardware microcomputer system from the real-world environment. They are the operating system, the support software, and the application software.

Two major software selections to be made are (1) choice of the disk operating system (DOS) to support the development task, and (2) choice of the language to implement the application. Although many different combinations of operating system and language are able to address the same types of applications, these choices frequently are critical since certain selections are clearly better than others for some types of applications. Of course these two choices are influenced significantly by the initial hardware selection, by personal biases, and by the user's level of expertise.

1.6.1 Disk operating systems

Our applications frequently involve software implementation of real-time signal processing algorithms, so we orient the discussions around this area. Real-time means different things to different people in computing. For our applications, consider real-time computing to be what is required of video arcade game machines. The microcomputer that serves as the central processing unit of the game machine must do all its computing and produce its results in a time frame that appears to the user to be instantaneous. The game would be far less fun if, each time you fired a missile, the processor required a minute or two to determine the missile's trajectory and establish whether or not it had collided with an enemy spacecraft.

A typical example of the need for real-time processing in biomedical computing is in the analysis of electrocardiograms in the intensive care unit of the hospital. In the typical television medical drama, the ailing patient is connected to a monitor that beeps every time the heart beats. If the monitor's microcomputer required a minute or two to do the complex pattern recognition required to recognize each valid heartbeat and then beeped a minute or so after the actual event, the device would be useless. The challenge in real-time computing is to develop programs to implement procedures (algorithms) that appear to occur instantaneously (actually a given task may take several milliseconds).

One DOS criterion to consider in the real-time environment is the compromise between flexibility and usability. Figure 1.11 is a plot illustrating this compromise for several general-purpose microcomputer DOSs that are potentially useful in developing solutions to many types of problems including real-time applications. As the axes are labeled, the most user-friendly, flexible DOS possible would plot at the origin. Any DOS with an optimal compromise between usability and flexibility would plot on the 45-degree line.

A DOS like Unix has a position on the left side of the graph because it is very flexible, thereby permitting the user to do any task characteristic of an operating system. That is, it provides the capability to maximally manipulate a hardware/software system with excellent control of input/output and other facilities. It also provides for multiple simultaneous users to do multiple simultaneous tasks (i.e., it is a multiuser, multitasking operating system). Because of this great flexibility, Unix requires considerable expertise to use all of its capabilities. Therefore it plots high on the graph.

On the other hand, the Macintosh is a hardware/software DOS designed for ease of use and for graphics-oriented applications. Developers of the Macintosh implemented the best user-to-machine interface that they could conceive of by sacrificing a great deal of the direct user control of the hardware system. The concept was to produce a personal computer that would be optimal for running application programs, not a computer to be used for writing new application programs. In fact Apple intended that the Lisa would be the development system for creating new Macintosh programs.

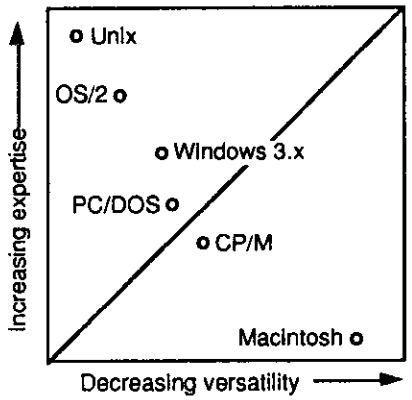


Figure 1.11 Disk operating systems–the compromise between DOS versatility and user expertise in real-time applications.

Without training, an individual can sit down and quickly learn to use a Macintosh because its operation is, by design, intuitive. The Macintosh DOS plots low and to the right because it is very user-friendly but cannot necessarily be used by the owner to solve any generalized problem. In fact, the Macintosh was the first personal computer in history that was sold without any language as part of the initial package. When the Macintosh was first introduced, no language was available for it, not even the ubiquitous BASIC that came free with almost every other computer at the time, including the original IBM PC.

The 8-bit operating system, CP/M (Control Program/Microprocessors), was a popular operating systems because it fell near the compromise line and represented a reasonable mixture of ease of use and flexibility. Also it could be implemented with limited memory and disk storage capability. CP/M was the most popular operating system on personal computers based on 8-bit microprocessors such as the Zilog Z80. PC DOS (or the generic MS DOS) was modeled after CP/M to fall near the compromise line. It became the most-used operating system on 16-bit personal computers, such as the IBM PC and its clones, that are based on the Intel 8086/8088 microprocessor or other 80x86 family members.

On the other hand, Unix is not popular on PCs because it requires a great deal of memory and hard disk storage to achieve its versatility. The latest Unix look-alike operating systems for the IBM PC and the Macintosh typically require significant memory and many megabytes of hard disk storage. This is compared to CP/M on an 8-bit system that normally was implemented in less than 20 kbytes of storage space and PC/DOS on an IBM PC that requires less than 100 kbytes.

At this writing, Unix is the workstation operating system of choice. For many applications, it may end up to be the DOS of choice. Indeed, Unix or a close clone

of it may ultimately provide the most accepted answer to the problem of linking PCs together through a local area network (LAN). By its very design, Unix provides multitasking, a feature necessary for LAN implementation. Incidentally, the fact that Unix is written in the C language gives it extraordinary transportability, facilitating its implementation on computers ranging from PCs to supercomputers.

For real-time digital filtering applications, Unix is not desirable because of its overhead compared to PC/DOS. In order to simultaneously serve multiple tasks, it must use up some computational speed. For the typical real-time problem, there is no significant speed to spare for this overhead. Real-time digital signal processing requires a single-user operating system. You must be able to extract the maximal performance from the computer and be able to manipulate its lowest level resources such as the hardware interrupt structure.

The trends for the future will be toward Macintosh-type operating systems such as Windows and OS/2. These user-friendly systems sacrifice a good part of the generalized computing power to the human-to-machine interface. Each DOS will be optimized for its intended application area and will be useful primarily for that area. Fully implemented versions of OS/2 will most likely require such large portions of the computing resources that they will have similar liabilities to those of Unix in the real-time digital signal processing environment.

Unfortunately there are no popular operating systems available that are specifically designed for such real-time applications as digital signal processing. A typical DOS is designed to serve the largest possible user base; that is, to be as general purpose as possible. In an IBM PC, the DOS is mated to firmware in the ROM BIOS (Basic Input/Output System) to provide a general, orderly way to access the system hardware. Use of high-level language calls to the BIOS to do a task such as display of graphics reduces software development time because assembly language is not required to deal directly with the specific integrated circuits that control the graphics. A program developed with high-level BIOS calls can also be easily transported to other computers with similar resources. However, the BIOS firmware is general purpose and has some inefficiencies. For example, to improve the speed of graphics refresh of the screen, you can use assembly language to bypass the BIOS and write directly to the display memory. However this speed comes at the cost of added development time and loss of transportability.

Of course, computers like the NEXT computer are attempting to address some of these issues. For example, the NEXT has a special shell for Unix designed to make it more user-friendly. It also includes a built-in digital signal processing (DSP) chip to facilitate implementation of signal processing applications.

1.6.2 Languages

Figure 1.12 shows a plot of the time required to write an application program as a function of run-time speed. This again is plotted for the case of real-time applications such as digital signal processing. The best language for this application area

time speed and the shortest development time. The diagonal line maps the best compromise language in terms of the run-time speed compared to the software design time necessary to implement an application. Of course there are other considerations for choosing a language, such as development cost and size of memory space available in an instrument. Also most of the languages plotted will not, by themselves, solve the majority of real-time problems, especially of the signal processing type.

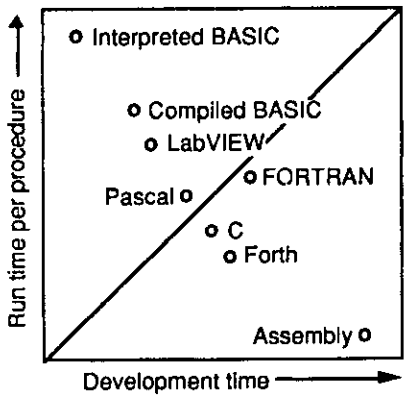


Figure 1.12 Languages–the compromise between development time and run-time speed in real-time applications.

The time to complete a procedure at run time is generally quite long for interpreted languages (that is, they are computationally slow). These interpreted languages like interpreted BASIC (Beginner's All-Purpose Symbolic Instruction Code) are generally not useful for real-time instrumentation applications, and they plot a great distance from the diagonal line.

The assembly language of a microprocessor is the language that can extract the greatest run-time performance because it provides for direct manipulation of the architecture of the processor. However it is also the most difficult language for writing programs, so it plots far from the optimal language line. Frequently, we must resort to this language in high-performance applications.

Other high-level languages such as FORTRAN and Pascal plot near the diagonal indicating that they are good compromises in terms of the trade-off between program development time and run time per procedure but do not usually produce code with enough run-time speed for real-time signal processing applications. Many applications are currently implemented by combining one of these languages with assembly language routines. FORTRAN was developed for solving equations (i.e., **FOR**mula **TRAN**slation) and Pascal was designed for teaching students structured programming techniques.

After several years of experience with a new microprocessor, software development companies are able to produce enhanced products. For example, modern versions of Pascal compilers developed for PCs have a much higher performance-to-price ratio than any Pascal compiler produced more than a decade ago.

Forth is useful for real-time applications, but it is a nontraditional, stack-oriented language so different from other programming languages that it takes some time for a person to become a skilled programmer. Documentation of programs is also difficult due to the flexibility of the language. Thus, a program developed in Forth typically is a one-person program. However, there are several small versions of the Forth compiler built into the same chip with a microprocessor. These implementations promote its use particularly for controller applications.

LabVIEW (National Instruments) is a visual computing language available only for the Macintosh that is optimized for laboratory applications. Programming is accomplished by interconnecting functional blocks (i.e., icons) that represent processes such as Fourier spectrum analysis or instrument simulators (i.e., virtual instruments). Thus, unlike traditional programming achieved by typing command statements, LabVIEW programming is purely graphical, a block diagram language. Although it is a relatively fast compiled language, LabVIEW is not optimized for real-time applications; its strengths lie particularly in the ability to acquire and process data in the laboratory environment.

The C language, which is used to develop modern versions of the Unix operating system, provides a significant improvement over assembly language for implementing most applications (Kernighan and Ritchie, 1978). It is the current language of choice for real-time programming. It is an excellent compromise between a low-level assembly language and a high-level language. C is standardized and structured. There are now several versions of commercial C++ compilers available for producing object-oriented software.

C programs are based on functions that can be evolved independently of one another and put together to implement an application. These functions are to software what black boxes are to hardware. If their I/O properties are carefully specified in advance, functions can be developed by many different software designers working on different aspects of the same project. These functions can then be linked together to implement the software design of a system.

Most important of all, C programs are transportable. By design, a program developed in C on one type of processor can be relatively easily transported to another. Embedded machine-specific functions such as those written in assembly language can be separated out and rewritten in the native code of a new architecture to which the program has been transported.

1.7 A LOOK TO THE FUTURE

As the microprocessor and its parent semiconductor technologies continue to evolve, the resulting devices will stimulate the development of many new types of medical instruments. We cannot even conceive of some of the possible applications now, because we cannot easily accept and start designing for the significant advances that will be made in computing in the next decade. With the 100-million-transistor microprocessor will come personal supercomputing. Only futurists can contemplate ways that we individually will be able to exploit such computing power. Even the nature of the microprocessor as we now know it might change more toward the architecture of the artificial neural network, which would lead to a whole new set of pattern recognition applications that may be more readily solvable than with today's microprocessors.

The choices of a laboratory computer, an operating system, and a language for a task must be done carefully. The IBM-compatible PC has emerged as a clear computer choice because of its widespread acceptance in the marketplace. The fact that so many PCs have been sold has produced many choices of hardware add-ons developed by numerous companies and also a wide diversity of software application programs and compilers. By default, IBM produced not only a hardware standard but also the clear-cut choice of the PC DOS operating system for the first decade of the use of this hardware. Although there are other choices now, DOS is still alive. Many will choose to continue using DOS for some time to come, adding to it a graphical user interface (GUI) such as that provided by Windows (Microsoft).

This leaves only the choice of a suitable language for your application area. My choice for biomedical instrumentation applications is C. In my view this is a clearly superior language for real-time computing, for instrumentation software design, and for other biomedical computing applications.

The hardware/software flexibility of the PC is permitting us to do research in areas that were previously too difficult, too expensive, or simply impossible. We have come a long way in biomedical computing since those innovators put together that first PC-like LINC almost three decades ago. Expect the PC and its descendants to stimulate truly amazing accomplishments in biomedical research in the next decade.

1.8 REFERENCES

Doumas, T. A., Tompkins, W. J., and Webster, J. G. 1982. An automatic calorie measuring device. *IEEE Frontiers of Eng. in Health Care*, **4**: 149–51.

Frisken-Gibson, S., Bach-y-Rita, P., Tompkins, W. J., and Webster, J. G. 1987. A 64-solenoid, 4-level fingertip search display for the blind. *IEEE Trans. Biomed. Eng.*, **BME-34**(12): 963–65.

Hamilton, P. S., and Tompkins, W. J. 1986. Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database. *IEEE Trans. Biomed. Eng.*, **BME-33**(12): 1157–65.

Hua, P., Woo, E. J., Webster, J. G., and Tompkins, W. J. 1991. Iterative reconstruction methods using regularization and optimal current patterns in electrical impedance tomography. *IEEE Trans. Medical Imaging*, **10**(4): 621–28.

Kaczmarek, K., Bach-y-Rita, P., Tompkins, W. J., and Webster, J. G. 1985. A tactile vision substitution system for the blind: computer-controlled partial image sequencing. *IEEE Trans. Biomed. Eng.*, **BME-32**(8):602–08.

Kernighan, B. W., and Ritchie, D. M. 1978. *The C programming language*. Englewood Cliffs, NJ: Prentice Hall.

Mehta, D., Tompkins, W. J., Webster, J. G., and Wertsch, J. J. 1989. Analysis of foot pressure waveforms. *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1487–88.

Pan, J. and Tompkins, W. J. 1985. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.*, **BME-32**(3): 230–36.

Pfister, C., Harrison, M. A., Hamilton, J. W., Tompkins, W. J., and Webster, J. G. 1989. Development of a 3-channel, 24-h ambulatory esophageal pressure monitor. *IEEE Trans. Biomed. Eng.*, **BME-36**(4): 487–90.

Tompkins, W. J. 1978. A portable microcomputer-based system for biomedical applications. *Biomed. Sci. Instrum.*, **14**: 61–66.

Tompkins, W. J. 1981. Portable microcomputer-based instrumentation. In H. S. Eden and M. Eden (eds.) *Microcomputers in Patient Care*. Park Ridge, NJ: Noyes Medical Publications, pp. 174–81.

Tompkins, W. J. 1985. Digital filter design using interactive graphics on the Macintosh. *Proc. of IEEE EMBS Annual Conf.*, pp. 722–26.

Tompkins, W. J. 1986. Biomedical computing using personal computers. *IEEE Engineering in Medicine and Biology Magazine*, **5**(3): 61–64.

Tompkins, W. J. 1988. Ambulatory monitoring. In J. G. Webster (ed.) *Encyclopedia of Medical Devices and Instrumentation*. New York: John Wiley, **1**:20–28.

Tompkins, W. J. and Webster, J. G. (eds.) 1981. *Design of Microcomputer-based Medical Instrumentation*. Englewood Cliffs, NJ: Prentice Hall.

Woo, E. J., Hua, P., Tompkins, W. J., and Webster, J. G. 1989. 32-electrode electrical impedance tomograph – software design and static images. *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 455–56.

Xue, Q. Z., Hu, Y. H. and Tompkins, W. J. 1992. Neural-network-based adaptive matched filtering for QRS detection. *IEEE Trans. Biomed. Eng.*, **BME-39**(4): 317–29.

Yorkey, T., Webster, J. G., and Tompkins, W. J. 1987. Comparing reconstruction algorithms for electrical impedance tomography. *IEEE Trans. Biomed. Eng.*, **BME-34**(11):843–52.

1.9 STUDY QUESTIONS

1.1 Compare operating systems for support in developing real-time programs. Explain the relative advantages and disadvantages of each for this type of application.

1.2 Explain the differences between interpreted, compiled, and integrated-environment compiled languages. Give examples of each type.

1.3 List two advantages of the C language for real-time instrumentation applications. Explain why they are important.

2

Electrocardiography

Willis J. Tompkins

One of the main techniques for diagnosing heart disease is based on the electrocardiogram (ECG). The electrocardiograph or ECG machine permits deduction of many electrical and mechanical defects of the heart by measuring ECGs, which are potentials measured on the body surface. With an ECG machine, you can determine the heart rate and other cardiac parameters.

2.1 BASIC ELECTROCARDIOGRAPHY

There are three basic techniques used in clinical electrocardiography. The most familiar is the standard clinical electrocardiogram. This is the test done in a physician's office in which 12 different potential differences called ECG leads are recorded from the body surface of a resting patient. A second approach uses another set of body surface potentials as inputs to a three-dimensional vector model of cardiac excitation. This produces a graphical view of the excitation of the heart called the vectorcardiogram (VCG). Finally, for long-term monitoring in the intensive care unit or on ambulatory patients, one or two ECG leads are monitored or recorded to look for life-threatening disturbances in the rhythm of the heartbeat. This approach is called arrhythmia analysis. Thus, the three basic techniques used in electrocardiography are:

1. Standard clinical ECG (12 leads)
2. VCG (3 orthogonal leads)
3. Monitoring ECG (1 or 2 leads)

Figure 2.1 shows the basic objective of electrocardiography. By looking at electrical signals recorded only on the body surface, a completely noninvasive procedure, cardiologists attempt to determine the functional state of the heart. Although the ECG is an electrical signal, changes in the mechanical state of the heart lead to changes in how the electrical excitation spreads over the surface of the heart,

thereby changing the body surface ECG. The study of cardiology is based on the recording of the ECGs of thousands of patients over many years and observing the relationships between various waveforms in the signal and different abnormalities. Thus clinical electrocardiography is largely empirical, based mostly on experiential knowledge. A cardiologist learns the meanings of the various parts of the ECG signal from experts who have learned from other experts.

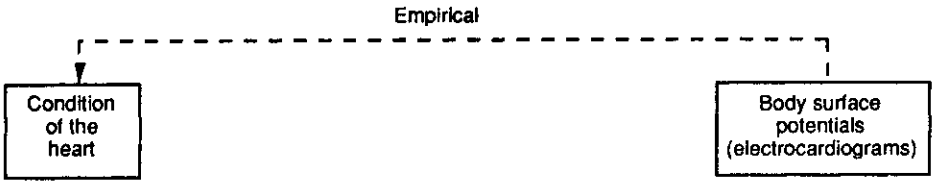


Figure 2.1 The object of electrocardiography is to deduce the electrical and mechanical condition of the heart by making noninvasive body surface potential measurements.

Figure 2.2 shows how the earliest ECGs were recorded by Einthoven at around the turn of the century. Vats of salt water provided the electrical connection to the body. The string galvanometer served as the measurement instrument for recording the ECG.

2.1.1 Electrodes

As time went on, metallic electrodes were developed to electrically connect to the body. An electrolyte, usually composed of salt solution in a gel, forms the electrical interface between the metal electrode and the skin. In the body, currents are produced by movement of ions whereas in a wire, currents are due to the movement of electrons. Electrode systems do the conversion of ionic currents to electron currents.

Conductive metals such as nickel-plated brass are used as ECG electrodes but they have a problem. The two electrodes necessary to acquire an ECG together with the electrolyte and the salt-filled torso act like a battery. A dc offset potential occurs across the electrodes that may be as large or larger than the peak ECG signal. A charge double layer (positive and negative ions separated by a distance) occurs in the electrolyte. Movement of the electrode such as that caused by motion of the patient disturbs this double layer and changes the dc offset. Since this offset potential is amplified about 1,000 times along with the ECG, small changes give rise to large baseline shifts in the output signal. An electrode that behaves in this way is called a polarizable electrode and is only useful for resting patients.

Figure 2.2 Early measurements of an ECG (about 1900) by Einthoven.

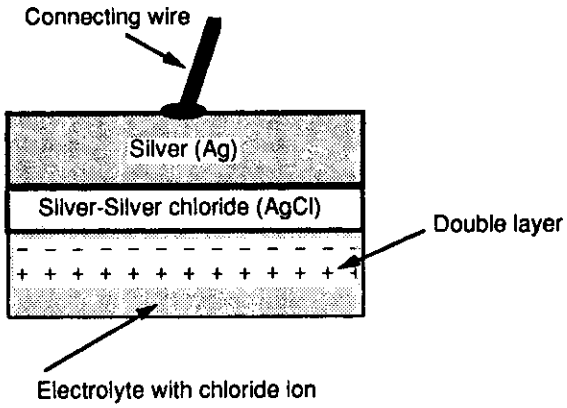


Figure 2.3 A silver-silver chloride ECG electrode. Many modern electrodes have electrolyte layers that are made of a firm gel which has adhesive properties. The firm gel minimizes the disturbance of the charge double layer.

The most-used material for electrodes these days is silver-silver chloride (Ag-AgCl) since it approximates a nonpolarizable electrode. Figure 2.3 shows such an electrode. This type of electrode has a very small offset potential. It has an AgCl layer deposited on an Ag plate. The chloride ions move in the body, in the electrolyte, and in the AgCl layer, where they get converted to electron flow in the Ag plate and in the connecting wire. This approach reduces the dc offset potential to a very small value compared to the peak ECG signal. Thus, movement of the electrode causes a much smaller baseline shift in the amplified ECG than that of a polarizable electrode.

2.1.2 The cardiac equivalent generator

Figure 2.4 shows how a physical model called a cardiac equivalent generator can be used to represent the cardiac electrical activity. The most popular physical model is a dipole current source that is represented mathematically as a time-varying vector which gives rise to the clinical vectorcardiogram (VCG). Einthoven postulated that the cardiac excitation could be modeled as a vector. He also realized that the limbs are like direct connections to points on the torso since the current fluxes set up inside the body by the dipole source flow primarily inside the thorax and do not flow significantly into the limbs. Thus he visualized a situation where electrodes could just as well have been connected to each of the shoulders and to a point near the navel had he not been restricted to using vats of saline.

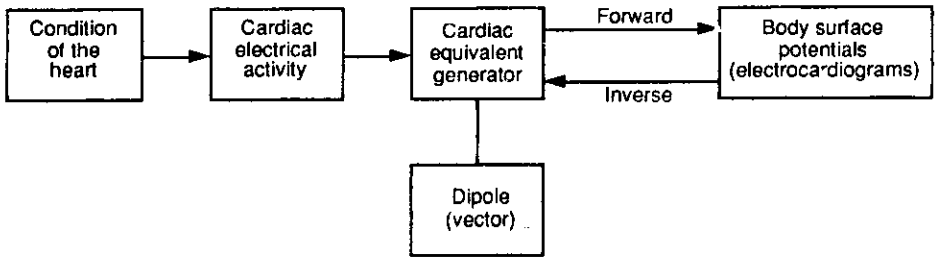


Figure 2.4 Both the electrical and mechanical conditions of the heart are involved in determining the characteristics of the spread of electrical activity over the surface of the heart. A model of this activity is called a cardiac equivalent generator.

Einthoven drew a triangle using as vertices the two shoulders and the navel and observed that the sides of the triangle were about the same length. This triangle, shown in Figure 2.5, has become known as the Einthoven equilateral triangle. If the vector representing the spread of cardiac excitation is known, then the potential difference measured between two limbs (i.e., two vertices of the triangle) is pro-

portional simply to the projection of the vector on the side of the triangle which connects the limbs. The figure shows the relationship between the Einthoven vector and each of the three frontal limb leads (leads I, II, and III). The positive signs show which connection goes to the positive input of the instrumentation amplifier for each lead.

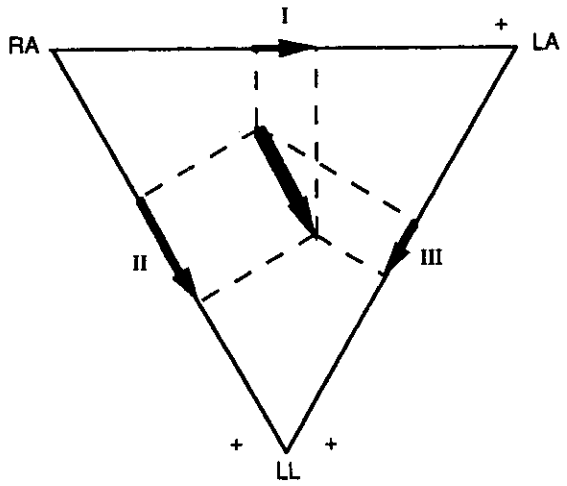


Figure 2.5 Einthoven equilateral triangle. RA and LA are the right and left arms and LL is the left leg.

A current dipole is a current source and a current sink separated by a distance. Since such a dipole has magnitude and direction which change throughout a heartbeat as the cells in the heart depolarize, this leads to the vector representation

$$\mathbf{p}(t) = p_x(t)\,\hat{\mathbf{x}} + p_y(t)\,\hat{\mathbf{y}} + p_z(t)\,\hat{\mathbf{z}} \tag{2.1}$$

where $\mathbf{p}(t)$ is the time-varying cardiac vector, $p_i(t)$ are the orthogonal components of the vector also called scalar leads, and $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}$ are unit vectors in the x, y, z directions.

A predominant VCG researcher in the 1950s named Frank shaped a plaster cast of a subject's body like the one shown in Figure 2.6, waterproofed it, and filled it with salt water. He placed a dipole source composed of a set of two electrodes on a stick in the torso model at the location of the heart. A current source supplied current to the electrodes which then produced current fluxes in the volume conductor. From electrodes embedded in the plaster, Frank measured the body surface potential distribution at many thoracic points resulting from the current

source. From the measurements in such a study, he found the geometrical transfer coefficients that relate the dipole source to each of the body surface potentials.

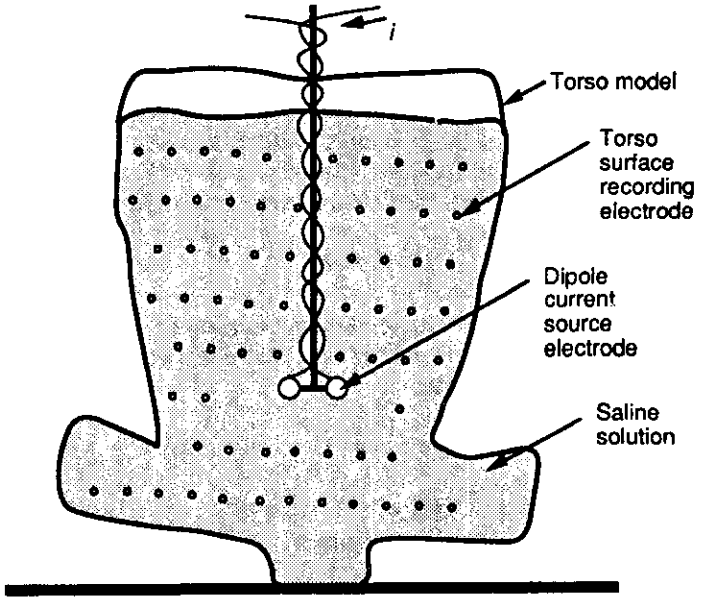


Figure 2.6 Torso model used to develop the Frank lead system for vectorcardiography.

Once the transfer coefficients are known, the forward problem of electrocardiography can be solved for any dipole source. The forward solution provides the potential at any arbitrary point on the body surface for a given cardiac dipole. Expressed mathematically,

$$v_n(t) = t_{nx}\, p_x(t) + t_{ny}\, p_y(t) + t_{nz}\, p_z(t) \tag{2.2}$$

This forward solution shows that the potential $v_n(t)$ (i.e., the ECG) at any point n on the body surface is given by the linear sum of the products of a set of transfer coefficients $[t_{ni}]$ unique to that point and the corresponding orthogonal dipole vector components $[p_i(t)]$. The ECGs are time varying as are the dipole components, while the transfer coefficients are only dependent on the thoracic geometry and inhomogeneities. Thus for a set of k body surface potentials (i.e., leads), there is a set of k equations that can be expressed in matrix form

$$\mathbf{V} = \mathbf{T} \times \mathbf{P} \tag{2.3}$$

where **V** is a $k \times 1$ vector representing the time-varying potentials, **T** is a $k \times 3$ matrix of transfer coefficients, which are fixed for a given individual, and **P** is the 3×1 time-varying heart vector.

Of course, the heart vector and transfer coefficients are unknown for a given individual. However if we had a way to compute this heart vector, we could use it in the solution of the forward problem and obtain the ECG for any body surface location. The approach to solving this problem is based on a physical model of the human torso. The model provides transfer coefficients that relate the potentials at many body surface points to the heart vector. With this information, we select three ECG leads that summarize the intrinsic characteristics of the desired abnormal ECG to simulate. Then we solve the inverse problem to find the cardiac dipole vector

$$\mathbf{P} = \mathbf{B} \times \mathbf{V} \tag{2.4}$$

where **B** is a $3 \times k$ matrix of lead coefficients that is directly derived from inverting the transfer coefficient matrix **T**. Thus, for the three heart vector components, there are three linear equations of the form

$$p_x(t) = b_{x1}\, v_1(t) + b_{x2}\, v_2(t) + \ldots + b_{xk}\, v_k(t) \tag{2.5}$$

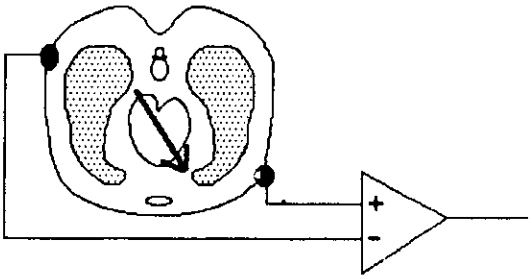
If we select k body surface ECG leads $\{v_1(t), v_2(t), \ldots, vk(t)\}$ for which the lead coefficients are known from the physical model of the human torso, we can solve the inverse problem and compute the time-varying heart vector. Once we have these dipole components, we solve the forward problem using Eq. (2.3) to compute the ECG for any point on the body surface.

2.1.3 Genesis of the ECG

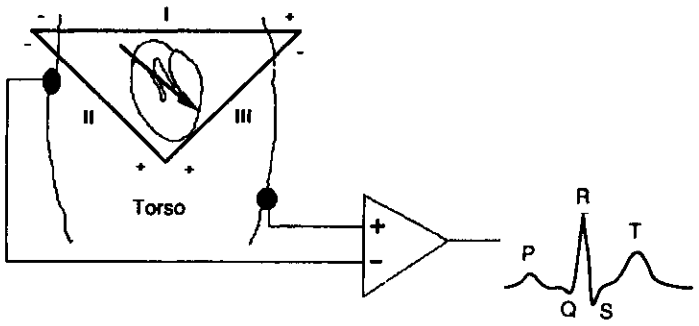
Figure 2.7 shows how an ECG is measured using electrodes attached to the body surface and connected to an instrumentation (ECG) amplifier. For the points in time that the vector points toward the electrode connected to the positive terminal of the amplifier, the output ECG signal will be positive-going. If it points to the negative electrode, the ECG will be negative. The time-varying motion of the cardiac vector produces the body surface ECG for one heartbeat with its characteristic P and T waves and QRS complex. Figure 2.8 shows a lead II recording for one heartbeat of a typical normal ECG.

Figure 2.9 illustrates how the cardiac spread of excitation represented by a vector at different points in time relates to the genesis of the body surface ECG for an amplifier configuration like the one in Figure 2.8. In Figure 2.9(a), the slow-moving depolarization of the atria which begins at the sinoatrial (SA) node produces the P wave. As Figure 2.9(b) shows, the signal is delayed in the atrioventricular (AV) node resulting in an isoelectric region after the P wave, then as the Purkinje system starts delivering the stimulus to the ventricular muscle, the onset of the Q wave occurs. In Figure 2.9(c), rapid depolarization of the ventricular muscle is de-

picted as a large, fast-moving vector which begins producing the R wave. Figure 2.9(d) illustrates that the maximal vector represents a point in time when most of the cells are depolarized, giving rise to the peak of the R wave. In Figure 2.9(e), the final phase of ventricular depolarization occurs as the excitation spreads toward the base of the ventricles (to the top in the picture) giving rise to the S wave.



(a)



(b)

Figure 2.7 Basic configuration for recording an electrocardiogram. Using electrodes attached to the body, the ECG is recorded with an instrumentation amplifier. (a) Transverse (top) view of a slice of the body showing the heart and lungs. (b) Frontal view showing electrodes connected in an approximate lead II configuration.

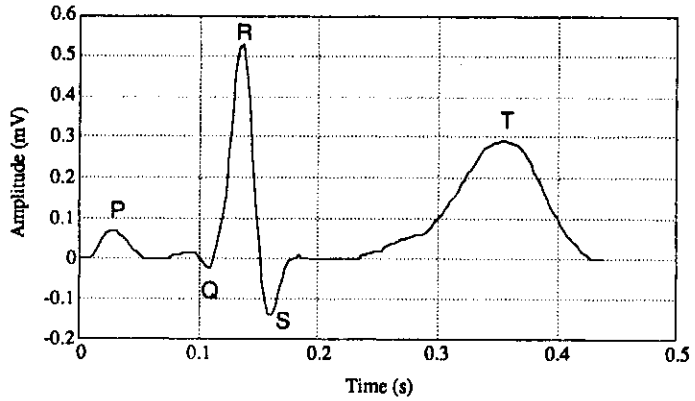


Figure 2.8 Electrocardiogram (ECG) for one normal heartbeat showing typical amplitudes and time durations for the P, QRS, and T waves.

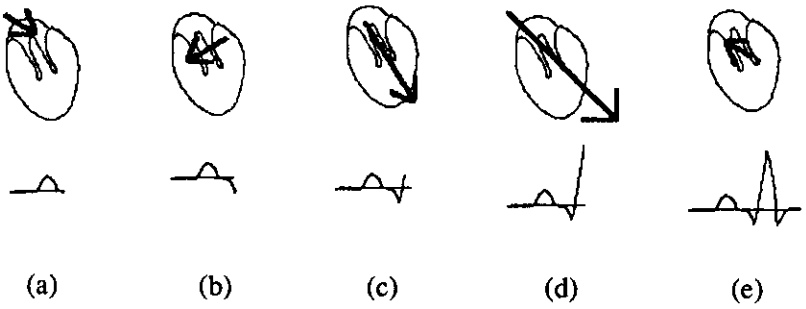


Figure 2.9 Relationship between the spread of cardiac electrical activation represented at various time instants by a summing vector (in the upper frames) and the genesis of the ECG (in the lower frames).

2.1.4 The standard limb leads

Figure 2.10 shows how we can view the potential differences between the limbs as ideal voltage sources since we make each voltage measurement using an instrumentation amplifier with a very high input impedance. It is clear that these three voltages form a closed measurement loop. From Kirchhoff's voltage law, the sum of the voltages around a loop equals zero. Thus

$$\text{II} - \text{I} - \text{III} = 0 \tag{2.6}$$

We can rewrite this equation to express any one of these leads in terms of the other two leads.

$$\text{II} = \text{I} + \text{III} \tag{2.7a}$$

$$\text{I} = \text{II} - \text{III} \tag{2.7b}$$

$$\text{III} = \text{II} - \text{I} \tag{2.7c}$$

It is thus clear that one of these voltages is completely redundant; we can measure any two and compute the third. In fact, that is exactly what modern ECG machines do. Most machines measure leads I and II and compute lead III. You might ask why we even bother with computing lead III; it is redundant so it has no new information not contained in leads I and II. For the answer to this question, we need to go back to Figure 2.1 and recall that cardiologists learned the relationships between diseases and ECGs by looking at a standard set of leads and relating the appearance of each to different abnormalities. Since these three leads were selected in the beginning, the appearance of each of them is important to the cardiologist.

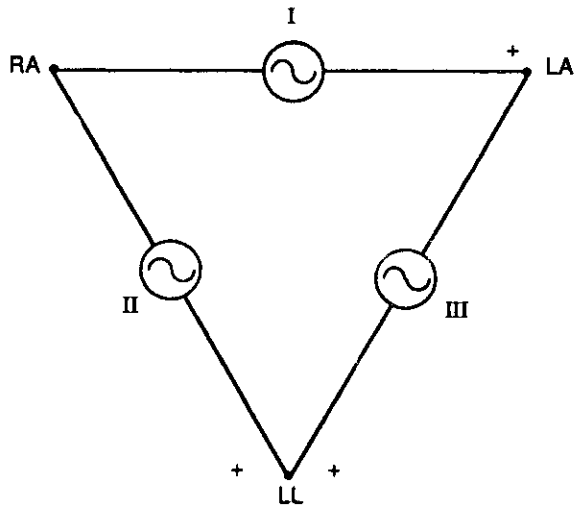


Figure 2.10 Leads I, II, and III are the potential differences between the limbs as indicated. RA and LA are the right and left arms and LL is the left leg.

2.1.5 The augmented limb leads

The early instrumentation had inadequate gain to produce large enough ECG traces

tude signals. In this case, the left arm signal, called augmented limb lead aVL, is measured using the average of the potentials on the other two limbs as a reference.

We can analyze this configuration using standard circuit theory. From the bottom left loop

$$i \times R + i \times R - \mathrm{II} = 0 \tag{2.8}$$

or

$$i \times R = \frac{\mathrm{II}}{2} \tag{2.9}$$

From the bottom right loop

$$-i \times R + \mathrm{III} + \mathrm{aVL} = 0 \tag{2.10}$$

or

$$\mathrm{aVL} = i \times R - \mathrm{III} \tag{2.11}$$

Combining Eqs. (2.9) and (2.11) gives

$$\mathrm{aVL} = \frac{\mathrm{II}}{2} - \mathrm{III} = \frac{\mathrm{II} - 2 \times \mathrm{III}}{2} \tag{2.12}$$

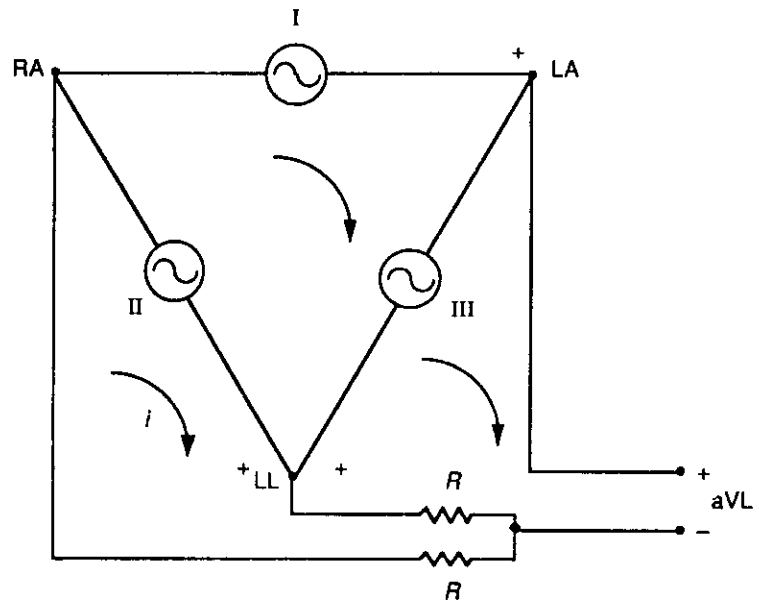


Figure 2.11 The augmented limb lead aVL is measured as shown.

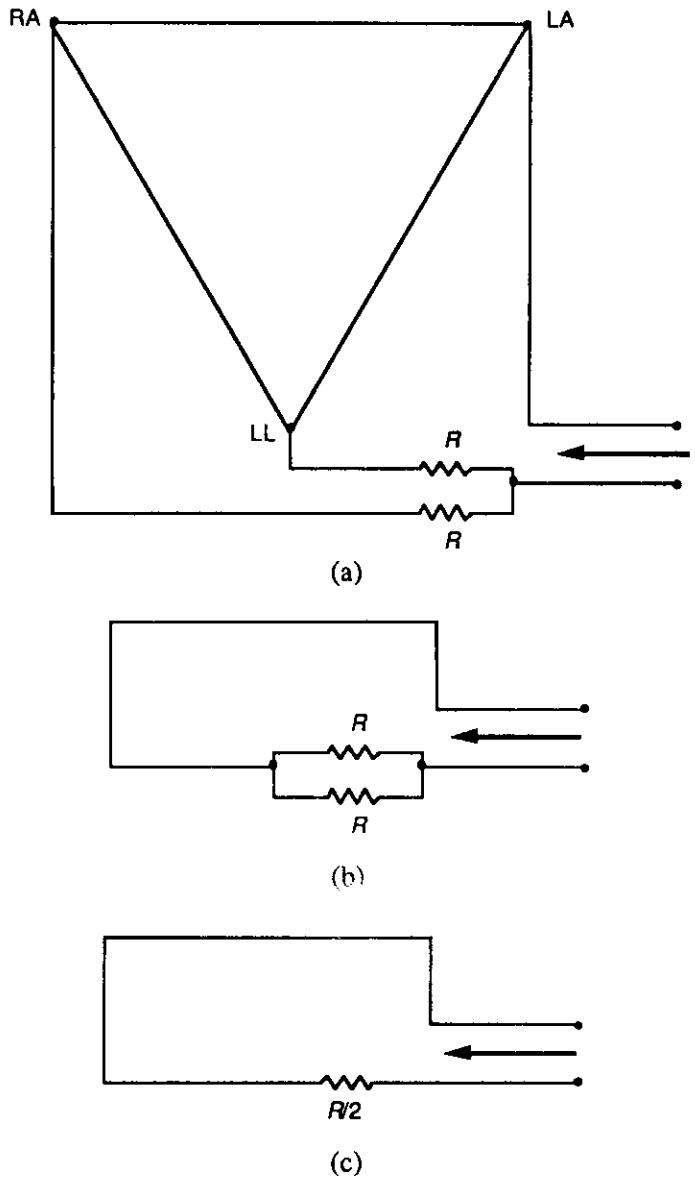


Figure 2.12 Determination of the Thévenin resistance for the aVL equivalent circuit. (a) All ideal voltage sources are shorted out. (b) This gives rise to the parallel combination of two equal resistors. (c) The Thévenin equivalent resistance thus has a value of $R/2$.

From the top center loop

$$\text{II} = \text{III} + \text{I} \tag{2.13}$$

Substituting gives

$$\text{aVL} = \frac{\text{III} + \text{I} - 2 \times \text{III}}{2} = \frac{\text{I} - \text{III}}{2} \tag{2.14}$$

This is the Thévenin equivalent voltage for the augmented lead aVL as an average of two of the frontal limb leads. It is clear that aVL is a redundant lead since it can be expressed in terms of two other leads. The other two augmented leads, aVR and aVF, similarly can both be expressed as functions of leads I and III. Thus here we find an additional three leads, all of which can be calculated from two of the frontal leads and thus are all redundant with no new real information. However due to the empirical nature of electrocardiology, the physician nonetheless still needs to see the appearance of these leads to facilitate the diagnosis.

Figure 2.12 shows how the Thévenin equivalent resistance is found by shorting out the ideal voltage sources and looking back from the output terminals.

Figure 2.13 illustrates that a recording system includes an additional resistor of a value equal to the Thévenin equivalent resistance connected to the positive input of the differential instrumentation amplifier. This balances the resistance at each input of the amplifier in order to ensure an optimal common mode rejection ratio (CMRR.

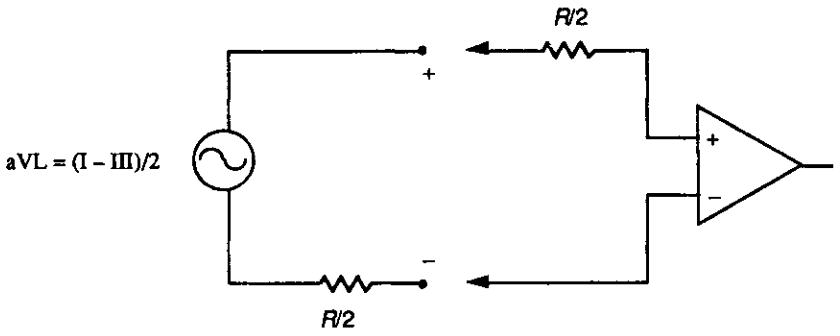


Figure 2.13 In a practical device for recording aVL, a resistance equal to the Thévenin equivalent resistance value of $R/2$ is added at the positive terminal of the instrumentation amplifier to balance the impedance on each input of the amplifier. This is done for optimal common mode performance.

Figure 2.14 shows how to solve vectorially for an augmented limb lead in terms of two of the standard limb leads. The limb leads are represented by vectors oriented in the directions of their corresponding triangle sides but centered at a common origin. To find aVL as in this example, we use the vectors of the two limb

leads that connect to the limb being measured, in this case, the left arm. We use lead I as one of the vectors to sum since its positive end connects to the left arm. We negate the vector for limb lead III (i.e., rotate it 180°) since its negative end connects to the left arm. Lead aVL is half the vector sum of leads I and –III [see Eq. (2.14)].

Figure 2.15 shows the complete set of vectors representing the frontal limb leads. From this depiction, you can quickly find all three augmented leads as functions of the frontal leads.

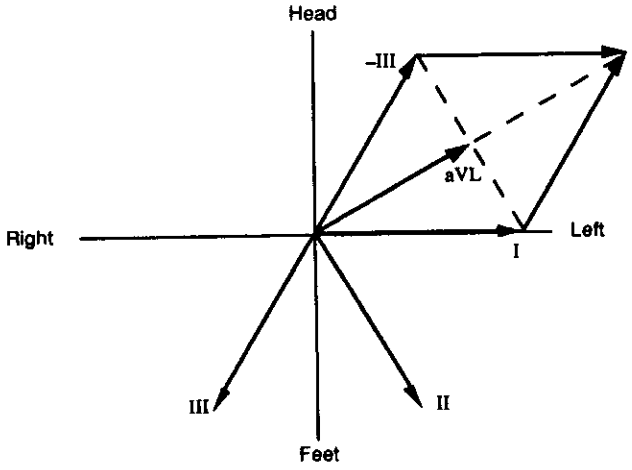


Figure 2.14 The vector graph solution for aVL in terms of leads I and III.

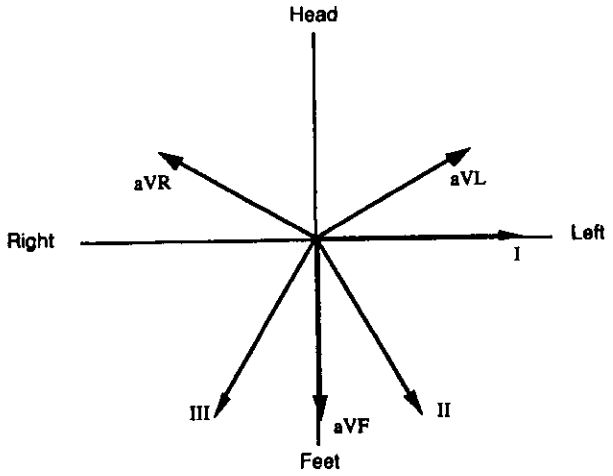


Figure 2.15 The vector relationships among all frontal plane leads.

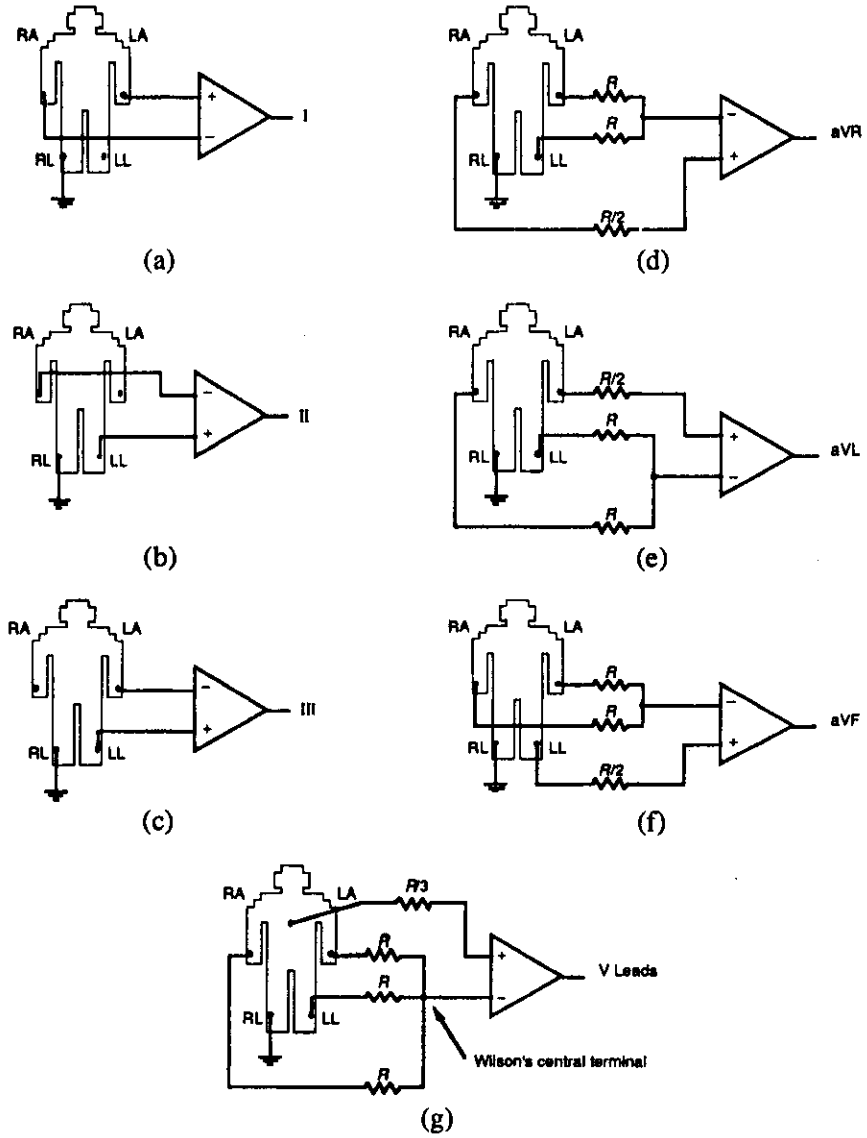


Figure 2.16 Standard 12-lead clinical electrocardiogram. (a) Lead I. (b) Lead II. (c) Lead III. Note the amplifier polarity for each of these limb leads. (d) aVR. (e) aVL. (f) aVF. These augmented leads require resistor networks which average two limb potentials while recording the third. (g) The six V leads are recorded referenced to Wilson's central terminal which is the average of all three limb potentials. Each of the six leads labeled V1–V6 are recorded from a different anatomical site on the chest.

2.2 ECG LEAD SYSTEMS

There are three basic lead systems used in cardiology. The most popular is the 12-lead approach, which defines the set of 12 potential differences that make up the standard clinical ECG. A second lead system designates the locations of electrodes for recording the VCG. Monitoring systems typically analyze one or two leads.

2.2.1 12-lead ECG

Figure 2.16 shows how the 12 leads of the standard clinical ECG are recorded, and Figure 2.17 shows the standard 12-lead ECG for a normal patient. The instrumentation amplifier is a special design for electrocardiography like the one shown in Figure 2.23. In modern microprocessor-based ECG machines, there are eight similar ECG amplifiers which simultaneously record leads I, II, and V1–V6. They then compute leads III, aVL, aVR, and aVF for the final report.

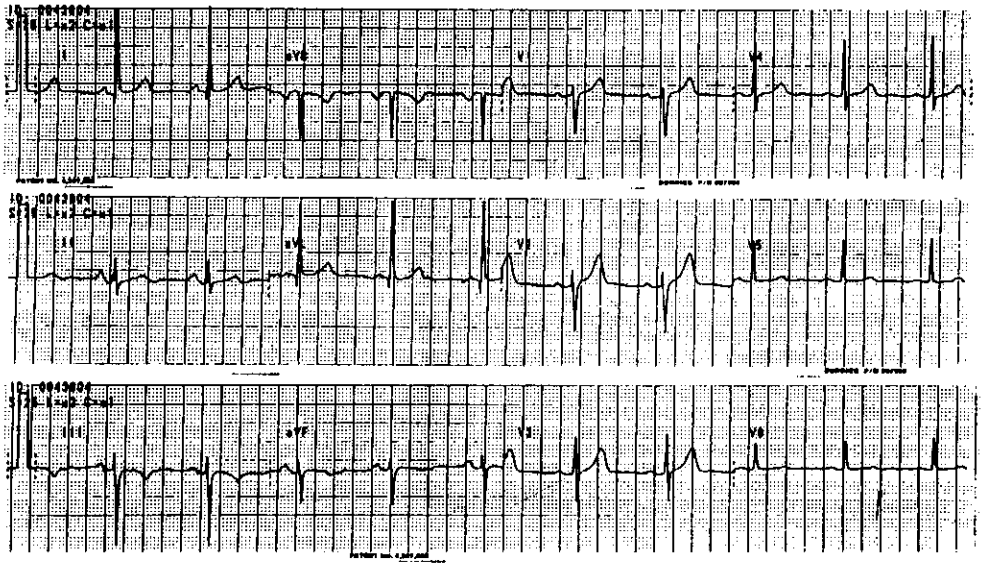


Figure 2.17 The 12-lead ECG of a normal male patient. Calibration pulses on the left side designate 1 mV. The recording speed is 25 mm/s. Each minor division is 1 mm, so the major divisions are 5 mm. Thus in lead I, the R-wave amplitude is about 1.1 mV and the time between beats is almost 1 s (i.e., heart rate is about 60 bpm).

2.2.2 The vectorcardiogram

Figure 2.18 illustrates the placement of electrodes for a Frank VCG lead system. Worldwide this is the most popular VCG lead system. Figure 2.19 shows how potentials are linearly combined with a resistor network to compute the three time-varying orthogonal scalar leads of the Frank lead system. Figure 2.20 is an IBM PC screen image of the VCG of a normal patient.

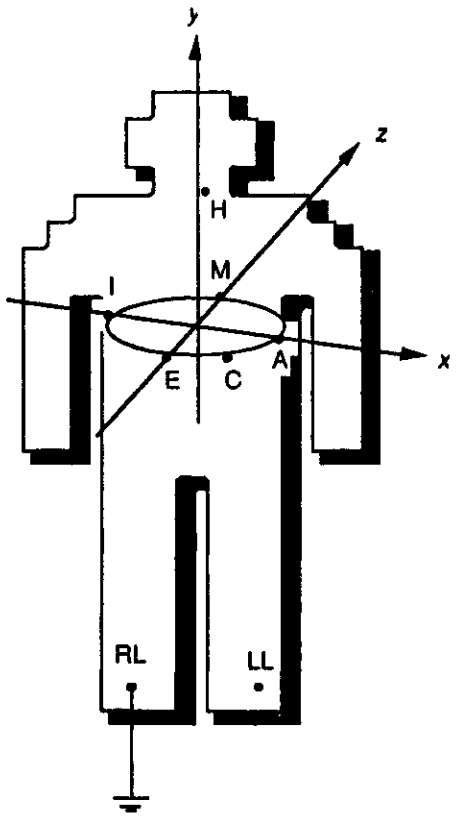
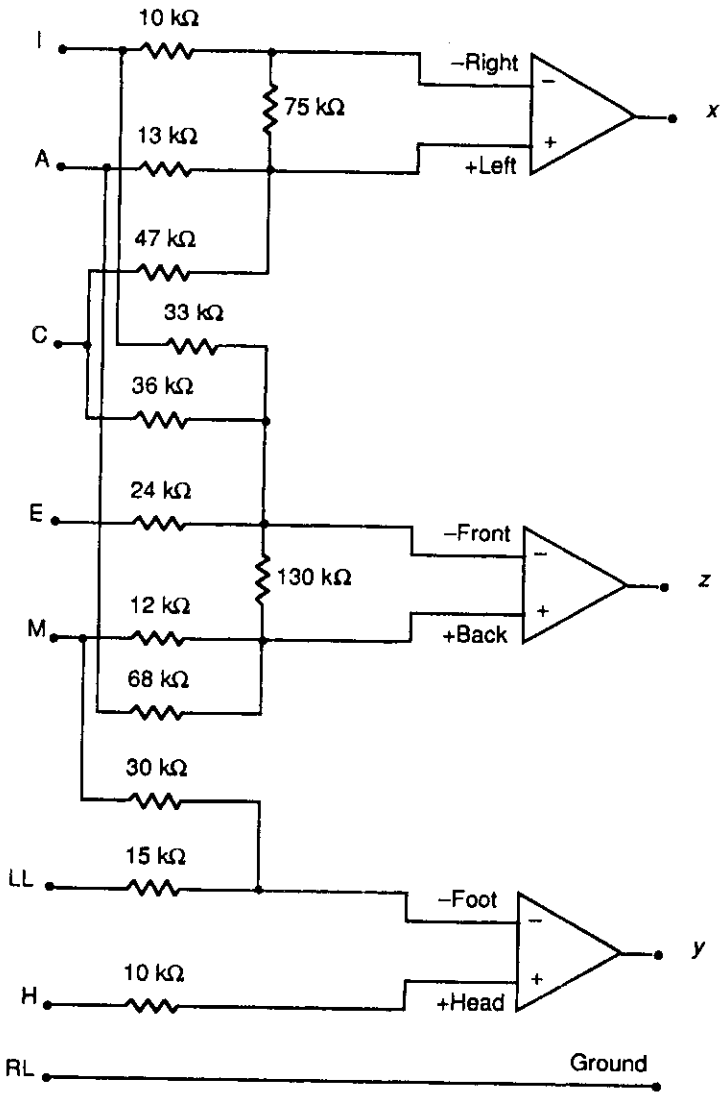


Figure 2.18 The electrode placement for the Frank VCG lead system.



(b)

Figure 2.19 The resistor network for combining body surface potentials to produce the three time-varying scalar leads of the Frank VCG lead system.

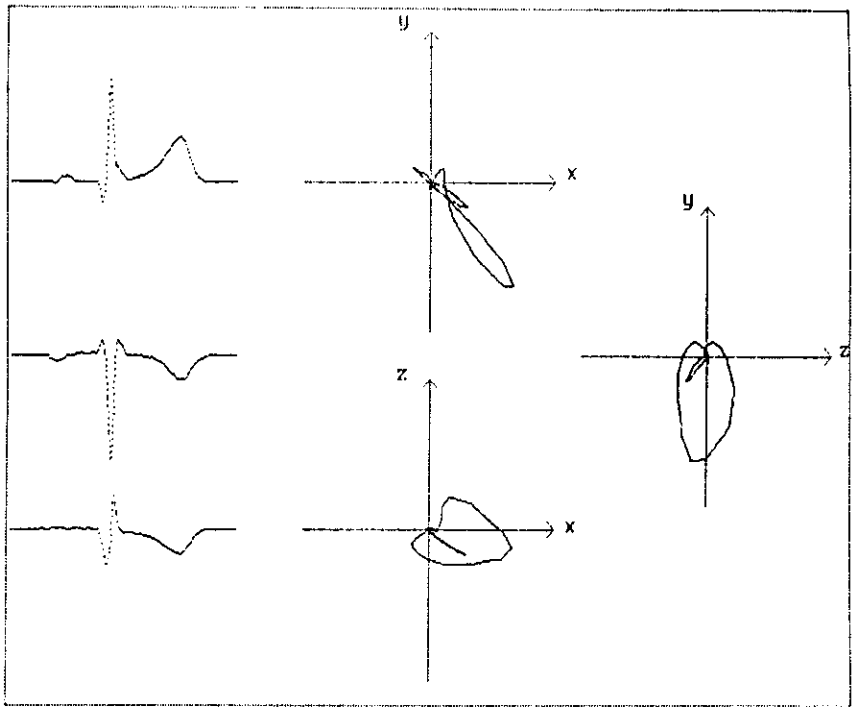


Figure 2.20 The vectorcardiogram of a normal male patient. The three time-varying scalar leads for one heartbeat are shown on the left and are the x, y, and z leads from top to bottom. In the top center is the frontal view of the tip of the vector as it moves throughout one complete heartbeat. In bottom center is a transverse view of the vector loop looking down from above the patient. On the far right is a left sagittal view looking toward the left side of the patient.

2.2.3 Monitoring lead systems

Monitoring applications do not use standard electrode positions but typically use two leads. Since the principal goal of these systems is to reliably recognize each heartbeat and perform rhythm analysis, electrodes are placed so that the primary ECG signal has a large R-wave amplitude. This ensures a high signal-to-noise ratio for beat detection. Since Lead II has a large peak amplitude for many patients, this lead is frequently recommended as the first choice of a primary lead by many manufacturers. A secondary lead with different electrode placements serves as a backup in case the primary lead develops problems such as loss of electrode contact.

2.3 ECG SIGNAL CHARACTERISTICS

Figure 2.21 shows three bandwidths used for different applications in electrocardiography (Tompkins and Webster, 1981). The clinical bandwidth used for recording the standard 12-lead ECG is 0.05–100 Hz. For monitoring applications, such as for intensive care patients and for ambulatory patients, the bandwidth is restricted to 0.5–50 Hz. In these environments, rhythm disturbances (i.e., arrhythmias) are principally of interest rather than subtle morphological changes in the waveforms. Thus the restricted bandwidth attenuates the higher frequency noise caused by muscle contractions (electromyographic or EMG noise) and the lower frequency noise caused by motion of the electrodes (baseline changes). A third bandwidth used for heart rate meters (cardiotachometers) maximizes the signal-to-noise ratio for detecting the QRS complex. Such a filter passes the frequencies of the QRS complex while rejecting noise including non-QRS waves in the signal such as the P and T waves. This filter helps to detect the QRS complexes but distorts the ECG so much that the appearance of the filtered signal is not clinically acceptable. One other application not shown extends the bandwidth up to 500 Hz in order to measure late potentials. These are small higher-frequency events that occur in the ECG following the QRS complex.

The peak amplitude of an ECG signal is in the range of 1 mV, so an ECG amplifier typically has a gain of about 1,000 in order to bring the peak signal into a range of about 1 V.

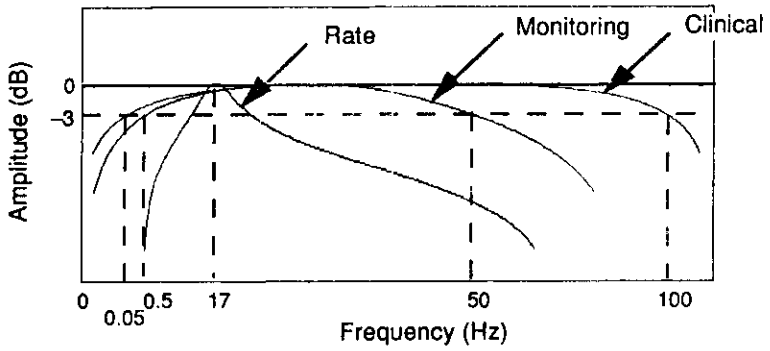


Figure 2.21 Bandwidths used in electrocardiography. The standard clinical bandwidth for the 12-lead clinical ECG is 0.05–100 Hz. Monitoring systems typically use a bandwidth of 0.5–50 Hz. Cardiotachometers for heart rate determination of subjects with predominantly normal beats use a simple bandpass filter centered at 17 Hz and with a Q of about 3 or 4.

2.4 LAB: ANALOG FILTERS, ECG AMPLIFIER, AND QRS DETECTOR*

In this laboratory you will study the characteristics of four types of analog filters: low-pass, high-pass, bandpass and bandstop. You will use these filters to build an ECG amplifier. Next you will study the application of a bandpass filter in a QRS detector circuit, which produces a pulse for each occurrence of a QRS complex. Note that you have to build all the circuits yourself.

2.4.1 Equipment

1. Dual trace oscilloscope
2. Signal generator
3. ECG electrodes
4. Chart recorder
5. Your ECG amplifier and QRS detection board
6. Your analog filter board

2.4.2 Background information

Low-pass filter/integrator

Figure 2.22(a) shows the circuit for a low-pass filter. The low-frequency gain, A_L, is given by

$$A_L = -\frac{R_2}{R_1} \tag{2.15}$$

The negative sign results because the op amp is in an inverting amplifier configuration. The high-corner frequency is given by

$$f_h = \frac{1}{2\pi R_2 C_1} \tag{2.16}$$

A low-pass filter acts like an integrator at high frequencies. The integrator output is given by

$$\begin{aligned} V_0 &= -\frac{1}{1 + j\omega R_1 C_1} V_i \\ &= -\frac{1}{R_1 C_1} \int V_i \, dt \end{aligned} \tag{2.17}$$

* Section 2.4 was written by Pradeep Tagare.

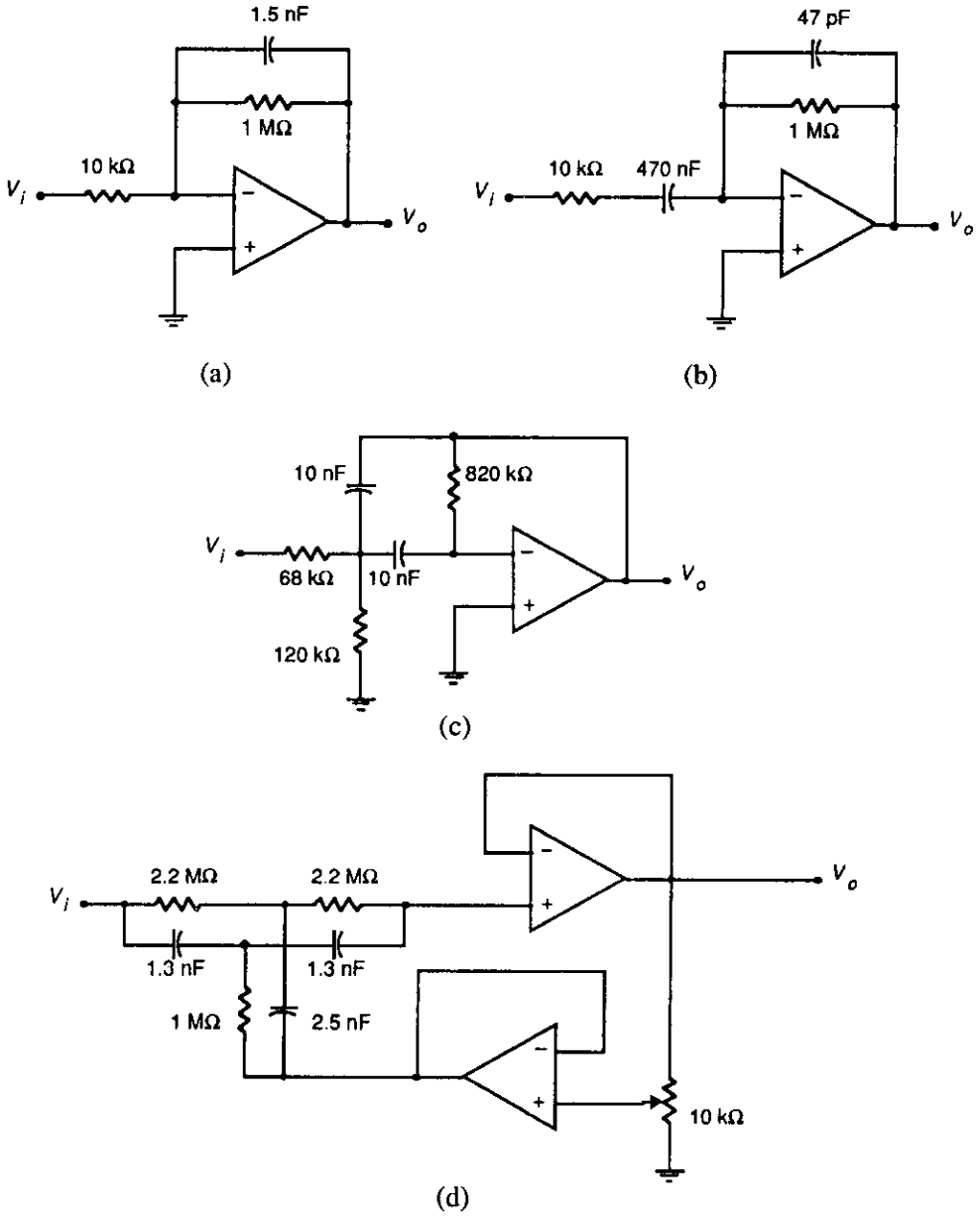


Figure 2.22 Analog filters. (a) Low-pass filter (integrator). (b) High-pass filter (differentiator). (c) Bandpass filter. (d) Bandstop (notch) filter.

This can be verified by observing the phase shift of the output with respect to the input. For a sinusoidal signal, the output is shifted by 90°

$$\int v \sin\omega t = -\frac{v}{\omega}\cos\omega t = \frac{v}{\omega}\sin(\omega t + \frac{\pi}{2}) \tag{2.18}$$

Thus the gain of the integrator falls at high frequencies. Also note that if $R2$ were not included in the integrator, the gain would become infinite at dc. Thus at dc the op amp dc bias current charges the integrating capacitor C_1 and saturates the amplifier.

High-pass filter/differentiator

In contrast to the low-pass filter which acts as an integrator at high frequencies, the high-pass filter acts like a differentiator at low frequencies. Referring to Figure 2.22(b), we get the high-frequency gain A_h and the low-corner frequency f_L as

$$A_h = -\frac{R_2}{R_1} \tag{2.19}$$

$$f_L = \frac{1}{2\pi R_1 C_1} \tag{2.20}$$

The differentiating behavior of the high-pass filter at low frequencies can be verified by deriving equations as was done for the integrator. Capacitor C_2 is added to improve the stability of the differentiator. The differentiator gain increases with frequency, up to the low-corner frequency.

Bandpass filter

The circuit we will use is illustrated in Figure 2.22(c). The gain of a bandpass filter is maximum at the center frequency and falls off on either side of the center frequency. The bandwidth of a bandpass filter is defined as the difference between the two corner frequencies. The Q of a bandpass filter is defined as

$$Q = \frac{\text{center frequency}}{\text{bandwidth}} \tag{2.21}$$

Bandstop/notch filter

Line frequency noise is a major source of interference. Sometimes a 60-Hz bandstop (notch) filter is used to reject this interference. Basically such a filter rejects one particular frequency while passing all other frequencies. Figure 2.22(d) shows

the bandstop filter that we will use. For the 60-Hz notch filter shown, the 60-Hz rejection factor is defined as

$$\text{60-Hz rejection factor} = \frac{\text{output voltage of the filter at 100 Hz}}{\text{output voltage of the filter at 60 Hz}} \quad (2.22)$$

for the same input voltage.

ECG amplifier

An ECG signal is usually in the range of 1 mV in magnitude and has frequency components from about 0.05–100 Hz. To process this signal, it has to be amplified. Figure 2.23 shows the circuit of an ECG amplifier. The typical characteristics of an ECG amplifier are high gain (about 1,000), 0.05–100 Hz frequency response, high input impedance, and low output impedance. Derivation of equations for the gain and frequency response are left as an exercise for the reader.

QRS detector

Figures 2.24 and 2.25 show the block diagram and complete schematic for the QRS detector. The QRS detector consists of the following five units:

1. QRS filter. The power spectrum of a normal ECG signal has the greatest signal-to-noise ratio at about 17 Hz. Therefore to detect the QRS complex, the ECG is passed through a bandpass filter with a center frequency of 17 Hz and a bandwidth of 6 Hz. This filter has a large amount of ringing in its output.
2. Half-wave rectifier. The filtered QRS is half-wave rectified, to be subsequently compared with a threshold voltage generated by the detector circuit.
3. Threshold circuit. The peak voltage of the rectified and filtered ECG is stored on a capacitor. A fraction of this voltage (threshold voltage) is compared with the filtered and rectified ECG output.
4. Comparator. The QRS pulse is detected when the threshold voltage is exceeded. The capacitor recharges to a new threshold voltage after every pulse. Hence a new threshold determined from the past history of the signal is generated after every pulse.
5. Monostable. A 200-ms pulse is generated for every QRS complex detected. This pulse drives a LED.

Some patients have a cardiac pacemaker. Since sharp pulses of the pacemaker can cause spurious QRS pulse detection, a circuit is often included to reject pacemaker pulses. The rejection is achieved by limiting the slew rate of the amplifier.

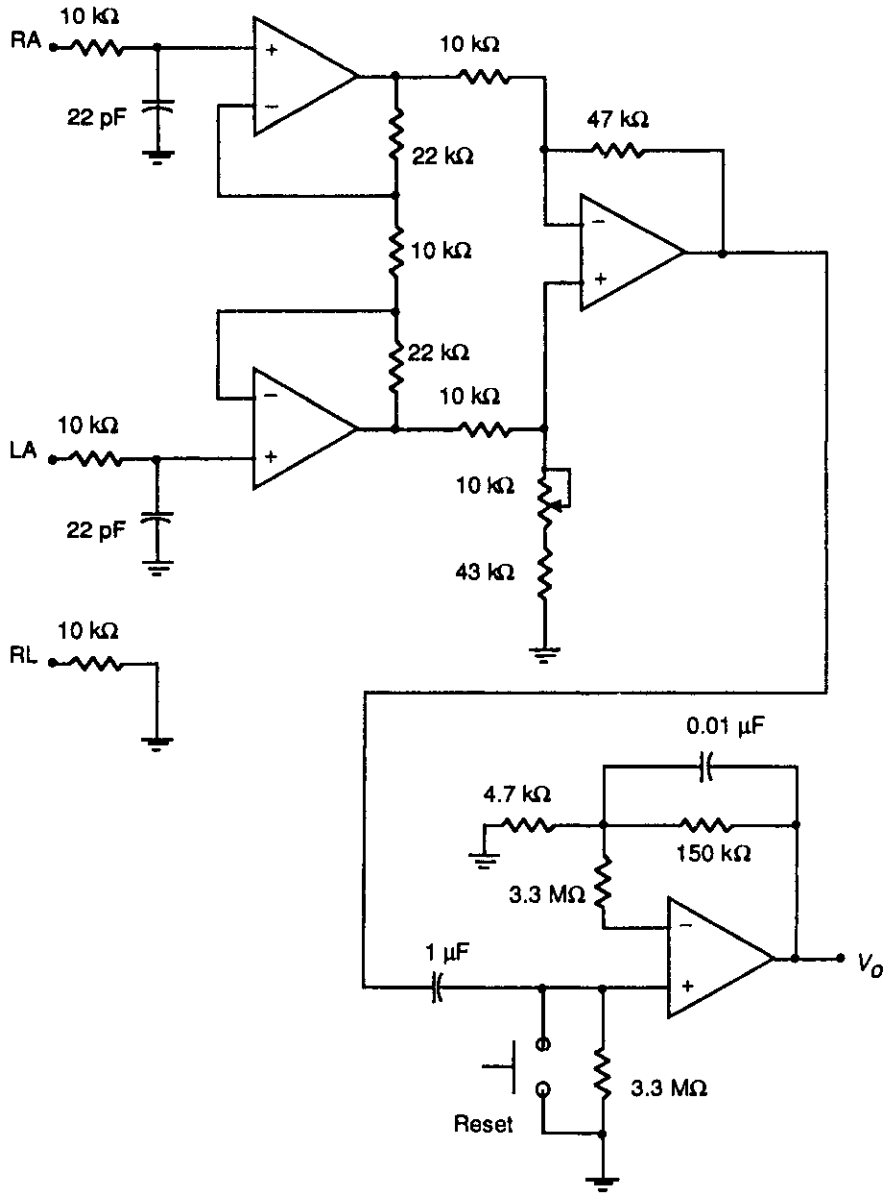


Figure 2.23 Circuit diagram of an ECG amplifier.

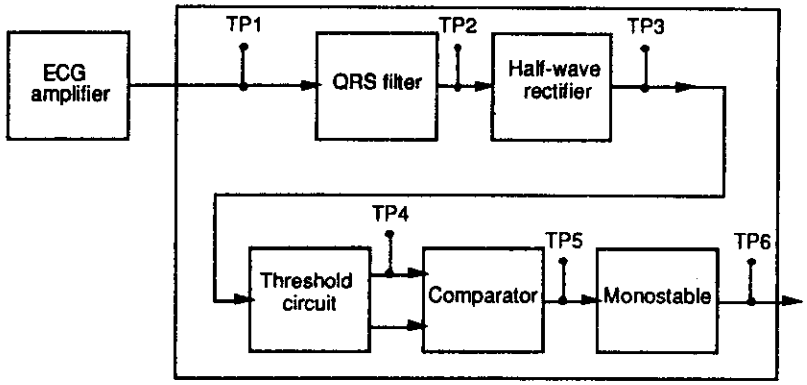


Figure 2.24 Block diagram of a QRS detector.

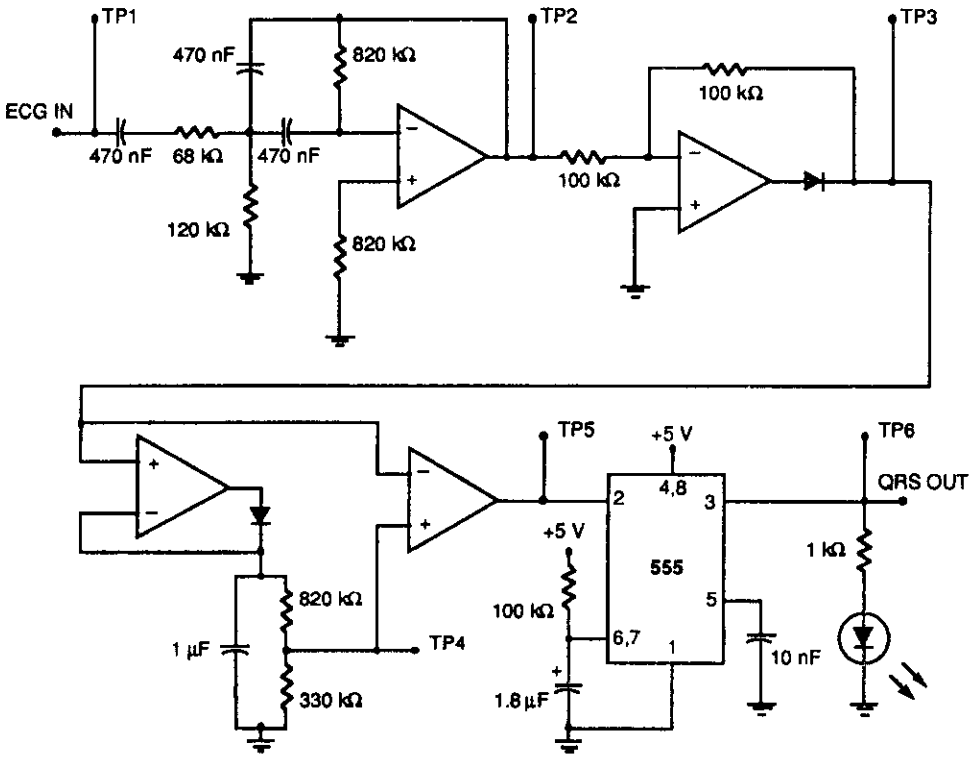


Figure 2.25 QRS detector circuit.

2.4.3 Experimental Procedure

Build all the circuits described above using the LM324 quad operational amplifier integrated circuit shown in Figure 2.26.

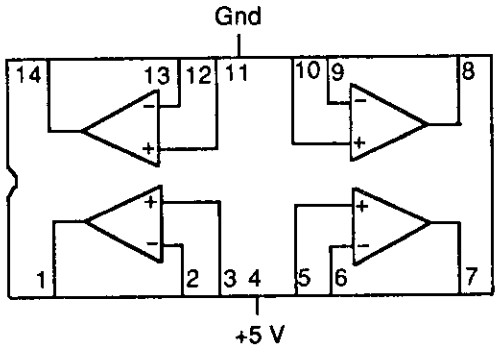


Figure 2.26 Pinout of the LM324 quad operational amplifier integrated circuit.

Low-pass filter

1. Turn on the power to the filter board. Feed a sinusoidal signal of the least possible amplitude generated by the signal generator at 10 Hz into the integrator input and observe both the input and the output on the oscilloscope. Calculate the gain.
2. Starting with a frequency of 10 Hz, increase the signal frequency in steps of 10 Hz up to 200 Hz and record the output at each frequency. You will use these values to plot a graph of the output voltage versus frequency. Next, find the generator frequency for which the output is 0.707-times that observed at 10 Hz. This is the –3 dB point or the high-corner frequency. Record this value.
3. Verify the operation of a low-pass filter as an integrator at high frequencies by observing the phase shift between the input and the output. Record the phase shift at the high-corner frequency.

High-pass filter

1. Feed a sinusoidal signal of the least possible amplitude generated by the signal generator at 200 Hz into the differentiator input and observe both the input and the output on the oscilloscope. Calculate the gain.
2. Starting with a frequency of 200 Hz, decrease the signal frequency in steps of 20 Hz to near dc and record the output at each frequency. You will use these values to plot a graph of the output voltage versus frequency. Next find the gen-

erator frequency for which the output is 0.707-times that observed at 200 Hz. This is the 3 dB point or the low-corner frequency. Record this value.
3. Verify the operation of a high-pass filter as a differentiator at low frequencies by observing the phase shift between the input and the output. Record the phase shift at the low-corner frequency. Another simple way to observe the differentiating behavior is to feed a 10-Hz square wave into the input and observe the spikes at the output.

Bandpass filter

For a 1-V p-p sinusoidal signal, vary the frequency from 10–150 Hz. Record the high- and low-corner frequencies. Find the center frequency and the passband gain of this filter.

Bandstop/notch filter

Feed a 1-V p-p 60-Hz sinusoidal signal into the filter, and measure the output voltage. Repeat the same for a 100-Hz sinusoid. Record results.

ECG amplifier

1. Connect LA and RA inputs of the amplifier to ground and observe the output. Adjust the 100-kΩ pot to null the offset voltage.
2. Connect LA and RA inputs to the signal high and the RL input to signal high (60 Hz) and RL to signal low. This is the common mode operation. Calculate the common mode gain.
3. Connect the LA input to the signal high (30 Hz) and the RA input to the signal low (through an attenuator to avoid saturation). This is the differential mode operation. Calculate the differential mode gain.
4. Find the frequency response of the amplifier.
5. Connect three electrodes to your body. Connect these electrodes to the amplifier inputs. Observe the amplifier output. If the signal is very noisy, try twisting the leads together. When you get a good signal, get a recording on the chart recorder.

QRS detector

1. Apply three ECG electrodes. Connect the electrodes to the input of the ECG amplifier board. Turn on the power to the board and observe the output of the ECG amplifier on the oscilloscope. Try pressing the electrodes if there is excessive noise.
2. Connect the output of the ECG amplifier to the input of the QRS detector board. Observe the following signals on the oscilloscope and then record them on a stripchart recorder with the ECG (TP1) on one channel and each of the other test signals (TP2–TP6) on the other channel. Use a reasonably fast paper speed (e.g., 25 mm/s).

Signals to be observed:

Test point	Signal
TP1	Your ECG
TP2	Filtered output
TP3	Rectified output
TP4	Comparator input
TP5	Comparator output
TP6	Monostable output

The LED should flash for every QRS pulse detected.

2.4.4 Lab report

1. Using equations described in the text, determine the values of A_L and f_h for the low-pass filter. Compare these values with the respective values obtained in the lab and account for any differences.
2. Plot the graph of the filter output voltage versus frequency. Show the –3 dB point on this graph.
3. What value of phase shift did you obtain for the low-pass filter?
4. Using equations described in the text, determine the values of A_H and f_L for the high-pass filter. Compare these values with the respective values obtained in the lab and account for any differences.
5. Plot the graph of the filter output voltage versus frequency. Show the –3 dB point on this graph.
6. What value of phase shift did you obtain for the high-pass filter?
7. For the bandpass amplifier, list the values that you got for the following:
   (a) center frequency
   (b) passband gain
   (c) bandwidth
   (d) Q
   Show all calculations.
8. What is the 60-Hz rejection factor for the bandstop filter you used?
9. What are the upper and lower –3 dB frequencies of your ECG amplifier? How do they compare with the theoretical values?
10. What is the gain of your ECG amplifier? How does it compare with the theoretical value?
11. What is the CMRR of your ECG amplifier?
12. How would you change the –3 dB frequencies of this amplifier?
13. Explain the waveforms you recorded on the chart recorder. Are these what you would expect to obtain?
14. Will the QRS detector used in this lab work for any person's ECG? Justify your answer.

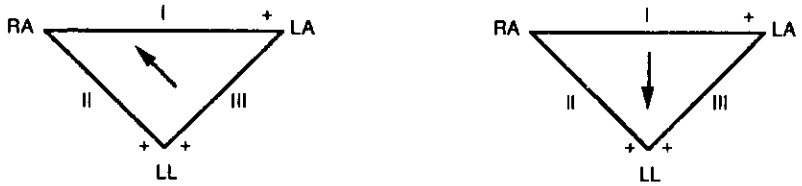
Include all chart recordings with your lab report and show calculations wherever propriate.

5 REFERENCES

mpkins, W. J. and Webster, J. G. (eds.) 1981. *Design of Microcomputer-based Medical Instrumentation*. Englewood Cliffs, NJ: Prentice-Hall.

6 STUDY QUESTIONS

What is a cardiac equivalent generator? How is it different from the actual cardiac electrical activity? Give two examples.
What is the vectorcardiogram and how is it recorded?
The heart vector of a patient is oriented as shown below at one instant of time. At this time, which of the frontal leads (I, II, and III) are positive-going for:



A certain microprocessor-based ECG machine samples and stores only leads I and II. What other standard leads can it compute from these two?
It is well known that all six frontal leads of the ECG can be expressed in terms of any two of them. Express the augmented lead at the right arm (i.e., aVR) in terms of leads I and II. Express Lead II in terms of aVF and aVL.
Is it possible to express lead V6 in terms of two other leads? Is there any way to calculate V6 from a larger set of leads?
There are four different bandwidths that are used in electrocardiography. Describe the principal applications for each of these bandwidths.
What is the frequency range of the standard 3-dB bandwidth used in (a) clinical electrocardiography, (b) electrocardiography monitoring applications such as in the intensive care unit? (c) Why are the clinical and monitoring bandwidths different?

0 A cardiologist records a patient's ECG on a machine that is suspected of being defective. She notices that the QRS complex of a normal patient's ECG has a lower peak-to-peak amplitude than the one recorded on a good machine. Explain what problems in instrument bandwidth might be causing this result.

1 A cardiologist notices that the T wave of a normal patient's ECG is distorted so that it looks like a biphasic sine wave instead of a unipolar wave. Explain what problems in instrument bandwidth might be causing this problem.

2 What is the electrode material that is best for recording the ECG from an ambulatory patient?

2.13 A cardiotachometer uses a bandpass filter to detect the QRS complex of the ECG. What its center frequency (in Hz)? How was this center frequency determined?

2.14 An engineer designs a cardiotachometer that senses the occurrence of a QRS complex wi a simple amplitude threshold. It malfunctions in two patients. (a) One patient's ECG h baseline drift and electromyographic noise. What ECG preprocessing step would provi the most effective improvement in the design for this case? (b) Another patient has a wave that is much larger than the QRS complex. This false triggers the thresholding circu What ECG preprocessing step would provide the most effective improvement in the desig for this case?

2.15 What is included in the design of an averaging cardiotachometer that prevents it from r sponding instantaneously to a heart rate change?

2.16 A typical modern microprocessor-based ECG machine samples and stores leads I, II, V V2, V3, V4, V5, and V6. From this set of leads, calculate (a) lead III, (b) augmente lead aVF.

3

Signal Conversion

David J. Beebe

The power of the computer to analyze and visually represent biomedical signals is of little use if the analog biomedical signal cannot be accurately captured and converted to a digital representation. This chapter discusses basic sampling theory and the fundamental hardware required in a typical signal conversion system. Section 3.1 discusses sampling basics in a theoretical way, and section 3.2 describes the circuits required to implement a real signal conversion system. We examine the overall system requirements for an ECG signal conversion system and discuss the possible errors involved in the conversion. We review digital-to-analog and analog-to-digital converters and other related circuits including amplifiers, sample-and-hold circuits, and analog multiplexers.

3.1 SAMPLING BASICS*

The whole concept of converting a continuous time signal to a discrete representation usable by a microprocessor, lies in the fact that we can represent a continuous time signal by its instantaneous amplitude values taken at periodic points in time. More important, we are able to reconstruct the original signal perfectly with just these sampled points. Such a concept is exploited in movies, where individual frames are snapshots of a continuously changing scene. When these individual frames are played back at a sufficiently fast rate, we are able to get an accurate representation of the original scene (Oppenheim and Willsky, 1983).

3.1.1 Sampling theorem

The *sampling theorem* initially developed by Shannon, when obeyed, guarantees that the original signal can be reconstructed from its samples without any loss of information. It states that, for a continuous bandlimited signal that contains no fre-

* Section 3.1 was written by Annie Foong.

quency components higher than f_c, the original signal can be completely recovered without distortion if it is sampled at a rate of at least $2 \times f_c$ samples/s. A sampling frequency f_s of twice the highest frequency *present* in a signal is called the Nyquist frequency.

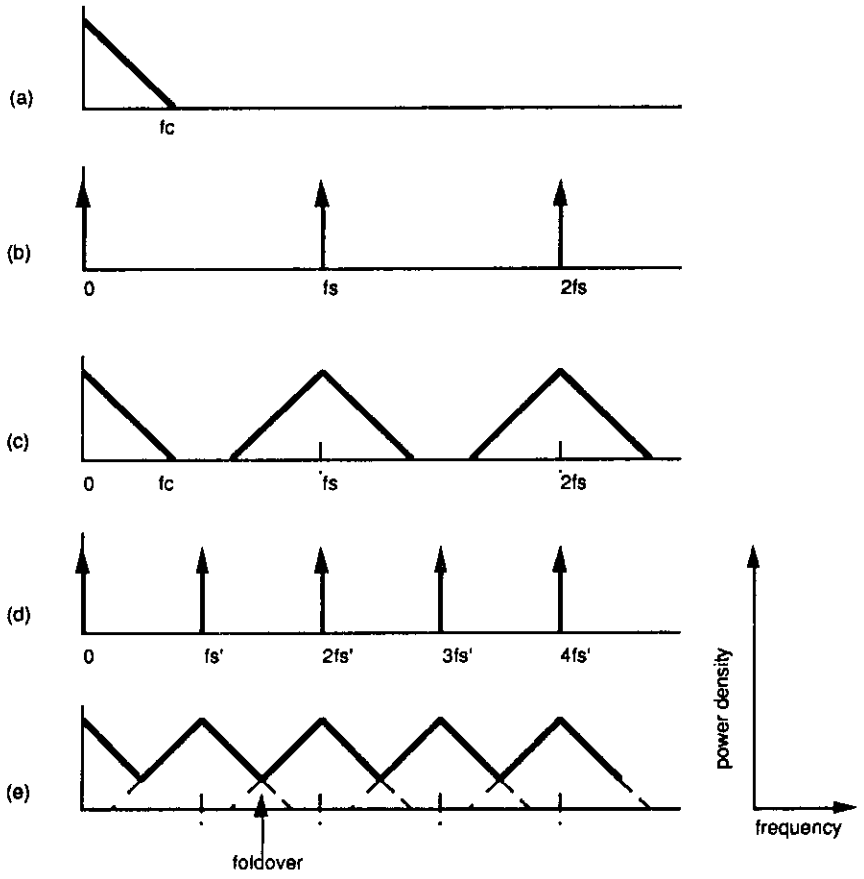


Figure 3.1 Effect in the frequency domain of sampling in the time domain. (a) Spectrum of original signal. (b) Spectrum of sampling function. (c) Spectrum of sampled signal with $f_s > 2f_c$. (d) Spectrum of sampling function with $f_s' < 2f_c$. (e) Spectrum of sampled signal with $f_s' < 2f_c$.

3.1.2 Aliasing, foldover, and other practical considerations

To gain more insight into the mechanics of sampling, we shall work in the frequency domain and deal with the spectra of signals. As illustrated in Figure 3.1(c),

if we set the sampling rate larger than $2 \times f_c$, the original signal can be recovered by placing a low-pass filter at the output of the sampler. If the sampling rate is too low, the situation in Figure 3.1(e) arises. Signals in the overlap areas are *dirtied* and cannot be recovered. This is known as *aliasing* where the higher frequencies are reflected into the lower frequency range (Oppenheim and Willsky, 1983).

Therefore, if we know in advance the highest frequency in the signal, we can theoretically establish the sampling rate at twice the highest frequency present. However, real-world signals are corrupted by noise that frequently contains higher frequency components than the signal itself. For example, if an ECG electrode is placed over a muscle, an unwanted electromyographic (EMG) signal may also be picked up (Cromwell et al., 1976). This problem is usually minimized by placing a low-pass filter at the sampler's input to keep out the unwanted frequencies.

However, nonideal filters can also prevent us from perfectly recovering the original signal. If the input filter is not ideal as in Figure 3.2(a), which is the case in practice, high frequencies may still slip through, and aliasing may still be present.

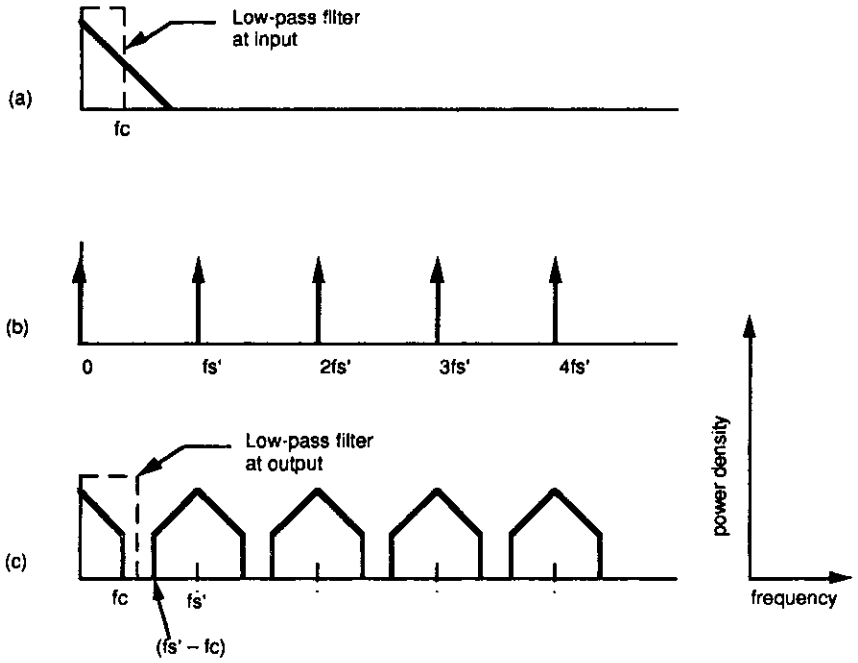


Figure 3.2 Effects of input and output filters. (a) Low-pass filter at input to remove high frequencies of signal. (b) Spectrum of sampling function. (c) No foldover present, low-pass filter at output to recover original signal.

Often ignored is the effect of the output filter. It can be seen in Figure 3.2(c) that, if the output filter is nonideal, the reconstructed signal may not be correct. In particular, the cutoff frequency of the output filter must be larger than f_c, but smaller than $(f_s' - f_c)$ so as not to include undesired components from the next sequence of spectra.

Finally, we may be limited by practical considerations not to set the sampling rate at Nyquist frequency even if we do know the highest frequency present in a signal. Most biomedical signals are in the low-frequency range. Higher sampling rates require more expensive hardware and larger storage space. Therefore, we usually tolerate an acceptable level of error in exchange for more practical sampling rates.

3.1.3 Examples of biomedical signals

Figure 3.3 gives the amplitudes and frequency bands of several human physiological signals.

Electroencephalogram (EEG)
Frequency range: dc–100 Hz (0.5–60 Hz)
Signal range: 15–100 mV

Electromyogram (EMG)
Frequency range: 10–200 Hz
Signal range: function of muscle activity and electrode placement

Electrocardiogram (ECG)
Frequency range: 0.05–100 Hz
Signal range: 10 μV(fetal), 5 mV(adult)

Heart rate
Frequency range: 45–200 beats/min

Blood pressure
Frequency range: dc–200 Hz (dc–60 Hz)
Signal range: 40–300 mm Hg (arterial); 0–15 mm Hg (venous)

Breathing rate
Frequency range: 12–40 breaths/min

Figure 3.3 Biomedical signals and ranges; major diagnostic range shown in brackets.

3.2 SIMPLE SIGNAL CONVERSION SYSTEMS

Biomedical signals come in all shapes and sizes. However, to capture and analyze these signals, the same general processing steps are required for all the signals. Figure 3.4 illustrates a general analog-to-digital (A/D) signal conversion system.

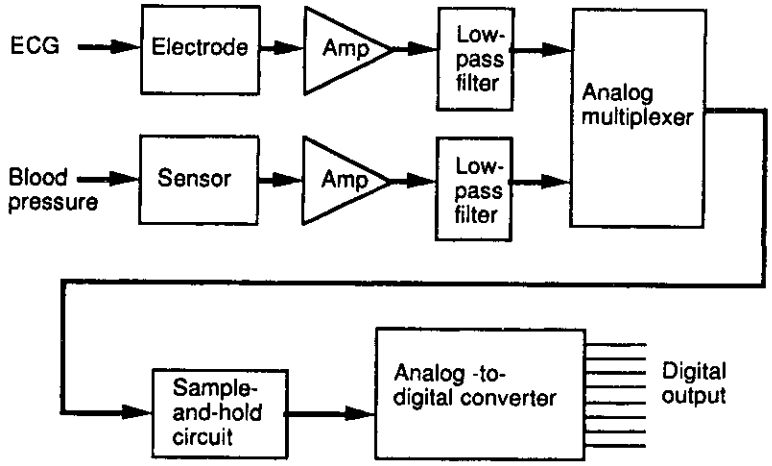


Figure 3.4 A typical analog-to-digital signal conversion system consists of sensors, amplifiers, multiplexers, a low-pass filter, a sample-and-hold circuit, and the A/D converter.

First the signal must be captured. If it is electrical in nature, a simple electrode can be used to pass the signal from the body to the signal conversion system. For other signals, a sensor is required to convert the biomedical signal into a voltage. The signal from the electrode or sensor is usually quite small in amplitude (e.g., the ECG ranges from 10 μV to 5 mV). Amplification is necessary to bring the amplitude of the signal into the range of the A/D converter. The amplification should be done as close to the signal source as possible to prevent any degradation of the signal. If there are several input signals to be converted, an analog multiplexer is needed to route each signal to the A/D converter. In order to minimize aliasing, a low-pass filter is often used to bandlimit the signal prior to sampling. A sample-and-hold circuit is required (except for very slowly changing signals) at the input to the A/D converter to hold the analog signal at a constant value during the conversion process. Finally, the A/D converter changes the analog voltage stored by the sample-and-hold circuit to a digital representation.

Now that a digital version of the biomedical signal has been obtained, what can it be used for? Often the digital information is stored in a memory device for later

digital signal processing algorithms commonly used for processing biomedical signals. In some cases this processing can be done in real time. Another possible use for the digital signal is in a control system. In this case the signal is processed by a computer and then fed back to the device to be controlled. Often the controller requires an analog signal, so a digital-to-analog (D/A) converter is needed. Figure 3.5 illustrates a general D/A conversion system. The analog output might be used to control the flow of gases in an anesthesia machine or the temperature in an incubator.

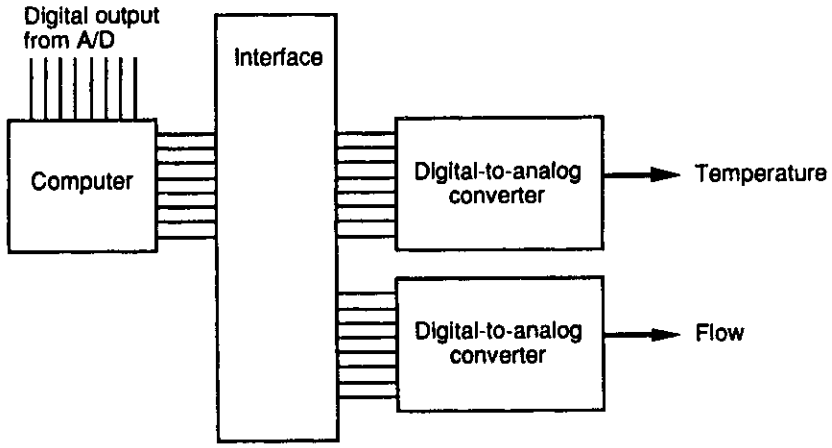


Figure 3.5 A typical digital-to-analog signal conversion system with a computer for processing the digital signal prior to conversion.

3.3 CONVERSION REQUIREMENTS FOR BIOMEDICAL SIGNALS

As discussed in section 3.1.3, biomedical signals have a variety of characteristics. The ultimate goal of any conversion system is to convert the biomedical signal to a digital representation with a minimal loss of information. The specifications for any conversion system are dependent on the signal characteristics and the application. In general, from section 3.1.3, one can see that biomedical signals are typically low frequency and low amplitude in nature. The following attributes should be considered when designing a conversion system: (1) accuracy, (2) sampling rate, (3) gain, (4) processing speed, (5) power consumption, and (6) size.

3.4 SIGNAL CONVERSION CIRCUITS

The digital representation of a continuous analog signal is discrete in both time (determined by the sampling rate) and amplitude (determined by the number of bits in a sampled data word). A variety of circuit configurations are available for converting signals between the analog and digital domains. Many of these are discussed in this chapter. Each method has its own advantages and shortcomings. The discussion here is limited to those techniques most commonly used in the conversion of biomedical signals. The D/A converter is discussed first since it often forms part of an A/D converter.

3.4.1 Converter characteristics

Before describing the details of the converter hardware, it is important to gain some knowledge of the basic terminology used in characterizing a converter's performance. One common method of examining the characteristics of a D/A converter (or an A/D converter) is by looking at its static and dynamic properties as described by Allen and Holberg (1987).

Static

For illustrative purposes a D/A converter is used, but the static errors discussed also apply to A/D converters. The ideal static behavior of a 3-bit D/A converter is shown in Figure 3.6. All the combinations of the digital input word are on the horizontal axis, while the analog output is on the vertical axis. The maximum analog output signal is 7/8 of the full scale reference voltage V_{ref}. For each unique digital input there should be a unique analog output. Any deviations from Figure 3.6 are known as static conversion errors. Static conversion errors can be divided into integral linearity, differential linearity, monotonicity, and resolution.

Integral linearity is the maximum deviation of the output of the converter from a straight line drawn from its ideal minimum to its ideal maximum. Integral linearity is often expressed in terms of a percentage of the full scale range or in terms of the least significant bit (LSB). Integral linearity can be further divided into absolute linearity, offset or zero error, full scale error, gain error, and monotonicity errors. Absolute linearity emphasizes the zero and full scale errors. The zero or offset error is the difference between the actual output and zero when the digital word for a zero output is applied. The full scale error is the difference between the actual and the ideal voltage when the digital word for a full scale output is applied. A gain error exists when the slope of the actual output is different from the slope of the ideal output. Figure 3.7(a) illustrates offset and gain errors. Monotonicity in a D/A converter means that as the digital input to the converter increases over its full scale range, the analog output never exhibits a decrease between one conversion step and the next. In other words, the slope of the output is never negative. Figure 3.7(b) shows the output of a converter that is nonmonotonic.

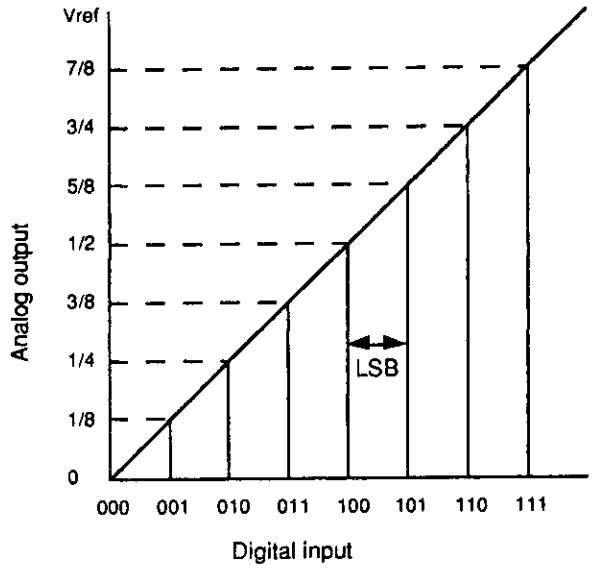


Figure 3.6 The ideal static behavior for a three-bit D/A converter. For each digital word there should be a unique analog signal.

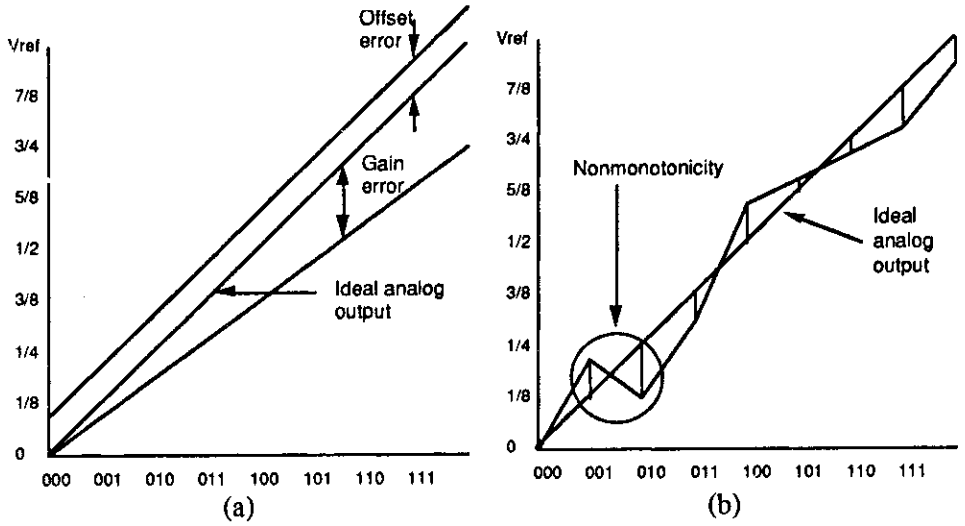


Figure 3.7 Digital-to-analog converter characteristics. (a) Gain and offset errors. (b) Monotonicity errors.

Differential linearity differs from integral linearity in that it is a measure of the separation between adjacent levels (Allen and Holberg, 1987). In other words, differential linearity measures bit-to-bit deviations from the ideal output step size of 1 LSB. Figure 3.8 illustrates the differences between integral and differential linearity.

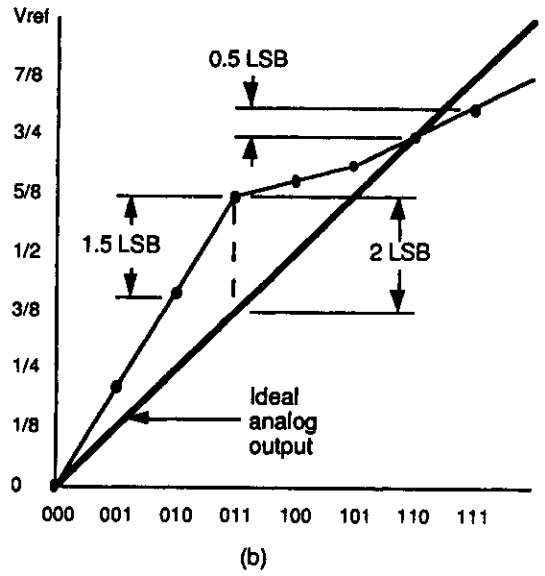


Figure 3.8 A D/A converter with ±2 LSB integral linearity and ±0.5 LSB differential linearity.

Resolution is defined as the smallest input digital code for which a unique analog output level is produced. Theoretically the resolution of an n-bit D/A converter is 2^n discrete analog output levels. In reality, the resolution is often less due to noise and component drift.

Dynamic

Settling time, for a D/A converter, is defined as the time it takes for the converter to respond to an input change. More explicitly, settling time is the time between the time a new digital signal is received at the converter input and the time when the output has reached its final value (within some specified tolerance). Many factors affect the settling time, so the conditions under which the settling time was found must be clearly stated if comparisons are to be made. Factors to note include the magnitude of the input change applied and the load on the output. Settling time in

D/A converters is important as it relates directly to the speed of the conversion. The output must settle sufficiently before it can be used in any further processing.

3.4.2 Digital-to-analog converters

The objective of the D/A converter is to construct an analog output signal for a given digital input. The first requirement for a D/A converter is an accurate voltage reference. Next the reference must be scaled to provide analog outputs at levels corresponding to each possible digital input. This is usually implemented using either voltage or charge scaling. Finally, the output can be interpolated to provide a smooth analog output.

Voltage reference

An accurate voltage reference is essential to the operation of a D/A converter. The analog output is derived from this reference voltage. Common reference voltage errors are either due to initial adjustments or generated by drifts with time and temperature. Two types of voltage references are used. One type uses the reverse breakdown voltage of a zener diode, while the other type derives its reference voltage from the extrapolated band-gap voltage of silicon. Temperature compensation is used in both cases. Scaling of this reference voltage is usually accomplished with passive components (resistors for voltage scaling and capacitors for charge scaling).

Voltage scaling

Voltage scaling of the reference voltage uses series resistors connected between the reference voltage and ground to selectively obtain discrete voltages between these limits. Figure 3.9 shows a simple voltage scaling 3-bit D/A converter. The digital input is decoded to select the corresponding output voltage. The voltage scaling structure is very regular and uses a small range of resistances. This is well suited to integrated circuit technology. Integrated circuit fabrication is best at making the same structure over and over. So while control over the absolute value of a resistor might be as high as 50 pecent, the relative accuracy can be as low as one percent (Allen and Holberg, 1987). That is, if we fabricate a set of resistors on one piece of silicon, each with a nominal value of 10 kΩ, the value of each might actually be 15 kΩ, but they will all be within one percent of each other in value.

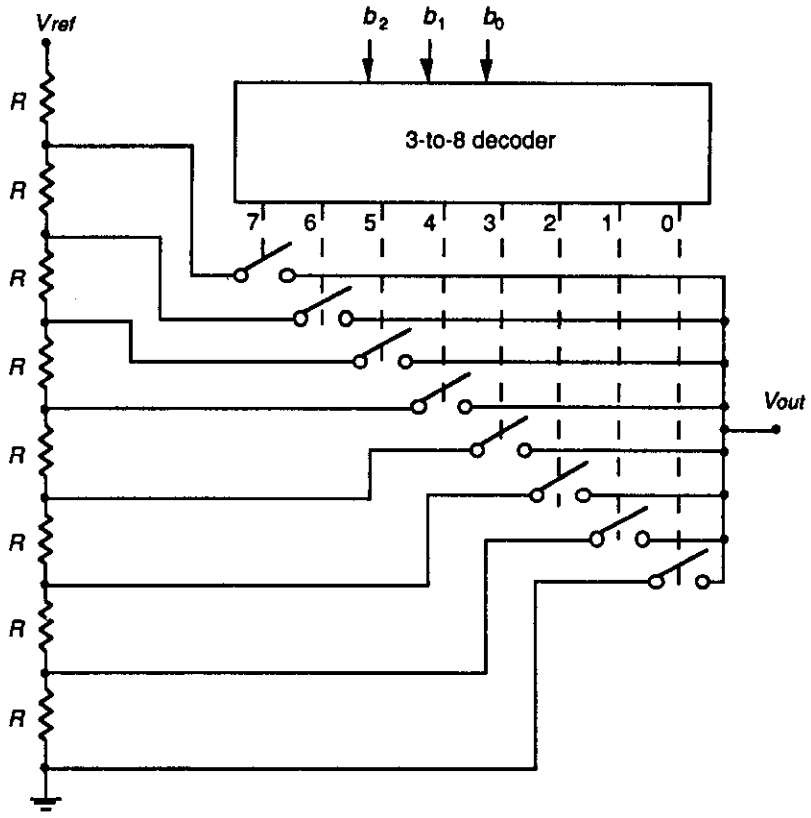


Figure 3.9 A simple voltage scaling D/A converter.

Charge scaling

Charge scaling D/A converters operate by doing binary division of the total charge applied to a capacitor array. Figure 3.10 shows a 3-bit charge scaling D/A converter. A two-phase clock is used. During phase 1, S_0 is closed and switches b_2, b_1, b_0 are closed shorting all the capacitors to ground. During phase 2 the capacitors associated with bits that are "1" are connected to V_{ref} and those with bits that are "0" are connected to ground. The output is valid only during phase 2. Equations (3.1) and (3.2) describe this operation. Note that the total capacitance is always $2C$ regardless of the number of bits in the word to be converted. The accuracy of charge scaling converters depends on the capacitor ratios. The error ratio for integrated circuit capacitors is frequently as low as 0.1 percent (Allen and Holberg, 1987).

$$C_{eq} = b_0 C + b_1 \frac{C}{2} + b_2 \frac{C}{4} \tag{3.1}$$

$$Vout = \frac{C_{eq}}{2C} \tag{3.2}$$

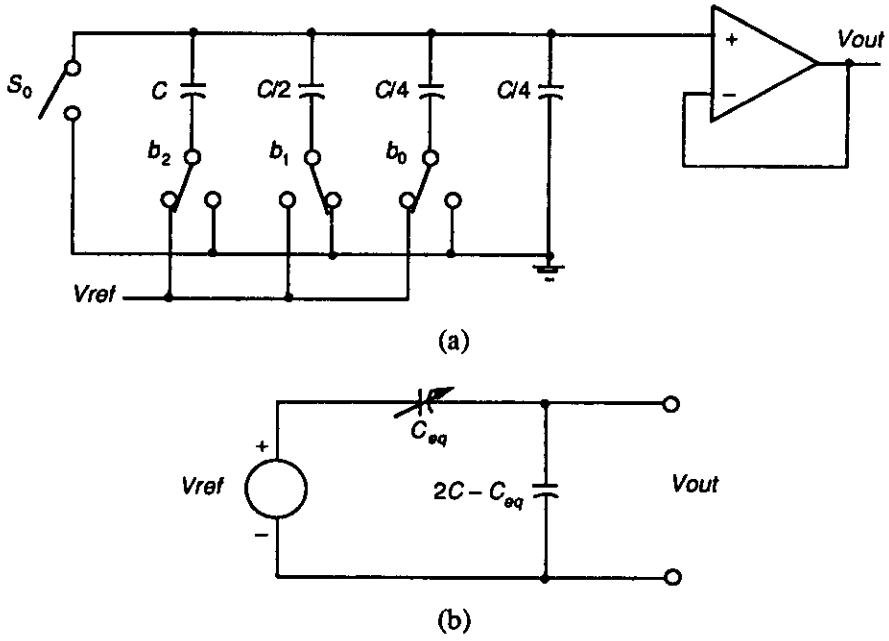


Figure 3.10 A 3-bit charge scaling D/A converter. a) Circuit with a binary 101 digital input. b) Equivalent circuit with any digital input.

Output interpolation

The outputs of simple D/A converters, such as those shown in Figures 3.9 and 3.10, are limited to discrete values. Interpolation techniques are often used to reconstruct an analog signal. Interpolation methods that can be easily implemented with electronic circuits include the following techniques: (1) zero-order hold or one-point, (2) linear or two-point, (3) bandlimited or low-pass (Tompkins and Webster, 1981).

3.4.3 Analog-to-digital converters

The objective of an A/D converter is to determine the output digital word for a given analog input. As mentioned previously, A/D converters often make use of D/A converters. Another method commonly used involves some form of integration or ramping. Finally, for high-speed sampling, a parallel or flash converter is used. In most converters a sample-and-hold circuit is needed at the input since it is not possible to convert a changing input signal. For very slowly changing signals, a sample-and-hold circuit is not always required. The errors associated with A/D converters are similar to those described in section 3.4.1 if the input and output definitions are interchanged.

Counter

The counter A/D converter increments a counter to build up the internal output one LSB at a time until it equals the analog input signal. A comparator stops the counter when the internal output has built up to the input signal level. At this point the count equals the digital output. The disadvantage of this scheme is a conversion time that varies with the level of the input signal. So for low-amplitude signals the conversion time can be fast, but if the signal amplitude doubles the conversion time will also double. Also, the accuracy of the conversion is subject to the error in the ramp generation.

Tracking

A variation of the counter A/D converter is the tracking A/D converter. While the counter converter resets its internal output to zero after each conversion, the internal output in the tracking converter continues to follow the analog input. Figure 3.11 illustrates this difference. By externally stopping the tracking A/D converter, it can be used as a sample-and-hold circuit with a digital output. Also by disabling the up or the down control, the tracking converter can be used to find the maximum or minimum value reached by the input signal over a given time period (Tompkins and Webster, 1988).

Dual slope

In a dual-slope converter, the analog input is integrated for a fixed interval of time (T_1). The length of this time is equal to the maximum count of the internal counter. The charge accumulated on the integrator's capacitor during this integration time is proportional to input voltage according to

$$Q = CV \tag{3.3}$$

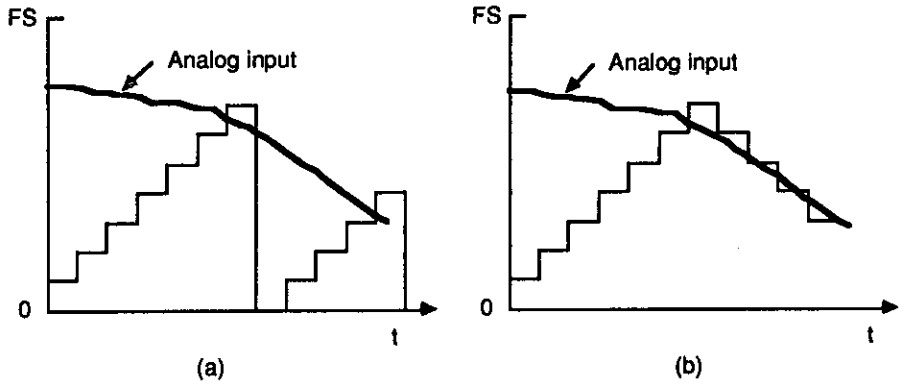


Figure 3.11 Internal outputs of the counter and tracking converters. a) The internal output of a counter A/D converter resets after each conversion. b) The internal output of the tracking A/D converter follows the analog input.

The slope of the integrator output is proportional to the amplitude of the analog input. After time T_1 the input to the integrator is switched to a negative reference voltage V_{ref}, thus the integrator integrates negatively with a constant slope. A counter counts the time t_2 that it takes for the integrator to reach zero. The charge gained by the integrator capacitor during T_1 must equal the charge lost during t_2

$$T_1 V_{in}(avg) = t_2 V_{ref} \tag{3.4}$$

Note that the ratio of t_2 to T_1 is also the ratio of the counter values

$$\frac{t_2}{T_1} = \frac{V_{in}}{V_{ref}} = \frac{\text{counter}}{\text{fixed count}} \tag{3.5}$$

So the count at the end of t_2 is equal to the digital output of the A/D converter. Figure 3.12 shows a block diagram of a dual-slope A/D converter and its associated waveforms. Note that the output of the dual-slope A/D converter is not a function of the slope of the integrator nor of the clock rate. As a result, this method of conversion is very accurate.

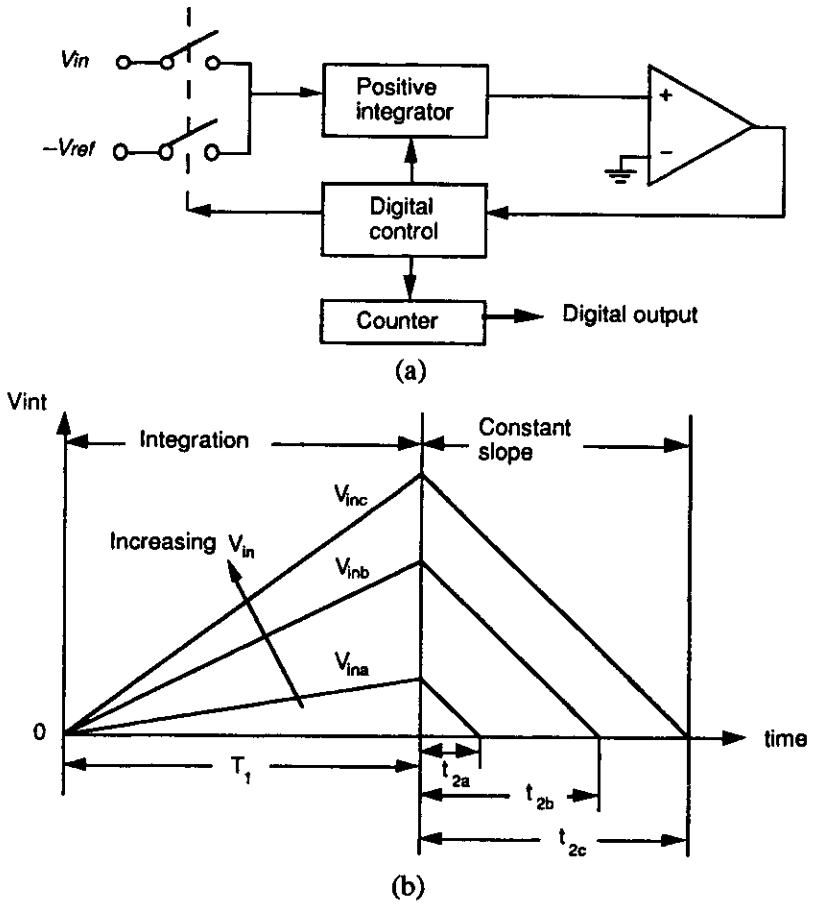


Figure 3.12 Dual-slope A/D converter. a) Block diagram. b) Waveforms illustrate the operation of the converter.

Successive approximation converter

The successive approximation converter uses a combination of voltage-scaling and charge-scaling D/A converters. Figure 3.13 shows a block diagram of a typical successive approximation A/D converter. It consists of a comparator, a D/A converter, and digital control logic. The conversion begins by sampling the analog signal to be converted. Next, the control logic assumes that the MSB is "1" and all other bits are "0". This digital word is applied to the D/A converter and an internal analog signal of 0.5 V_{ref} is generated. The comparator is now used to compare this generated analog signal to the analog input signal. If the comparator output is high, then the MSB is indeed "1". If the comparator output is "0", then the MSB is

changed to "0". At this point the MSB has been determined. This process is repeated for each remaining bit in order.

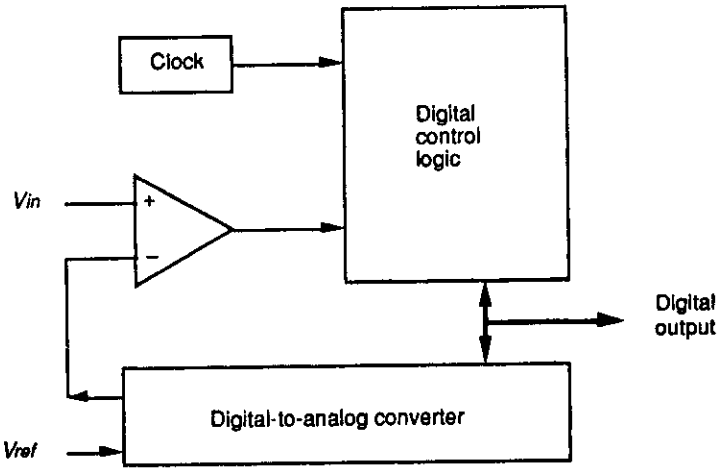


Figure 3.13 Block diagram of a typical successive approximation A/D converter.

Figure 3.14 shows the possible conversion paths for a 3-bit converter. Note that the number of clock cycles required to convert an n-bit word is n.

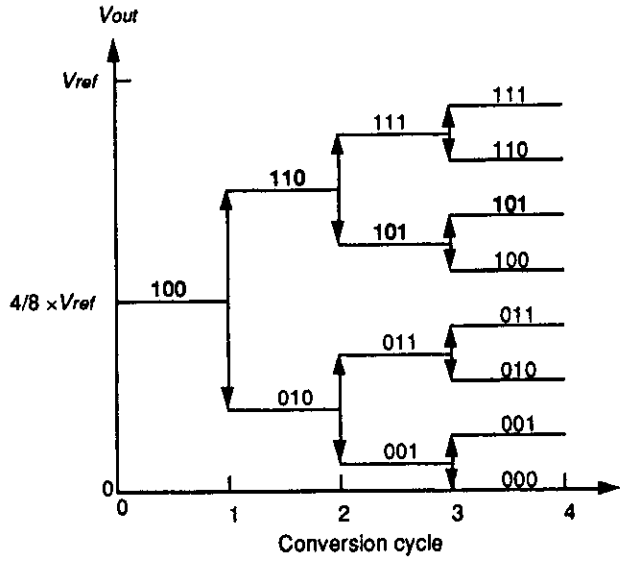


Figure 3.14 The successive approximation process. The path for an analog input equal to $5/8 \times Vref$ is shown in bold.

Parallel or flash

For very high speed conversions a parallel or flash type converter is used. The ultimate conversion speed is one clock cycle, which would consist of a setup and convert phase. In this type of converter the sample time is often the limiting factor for speed. The operation is straightforward and it is illustrated in Figure 3.15. To convert an n-bit word, $2^n - 1$ comparators are required.

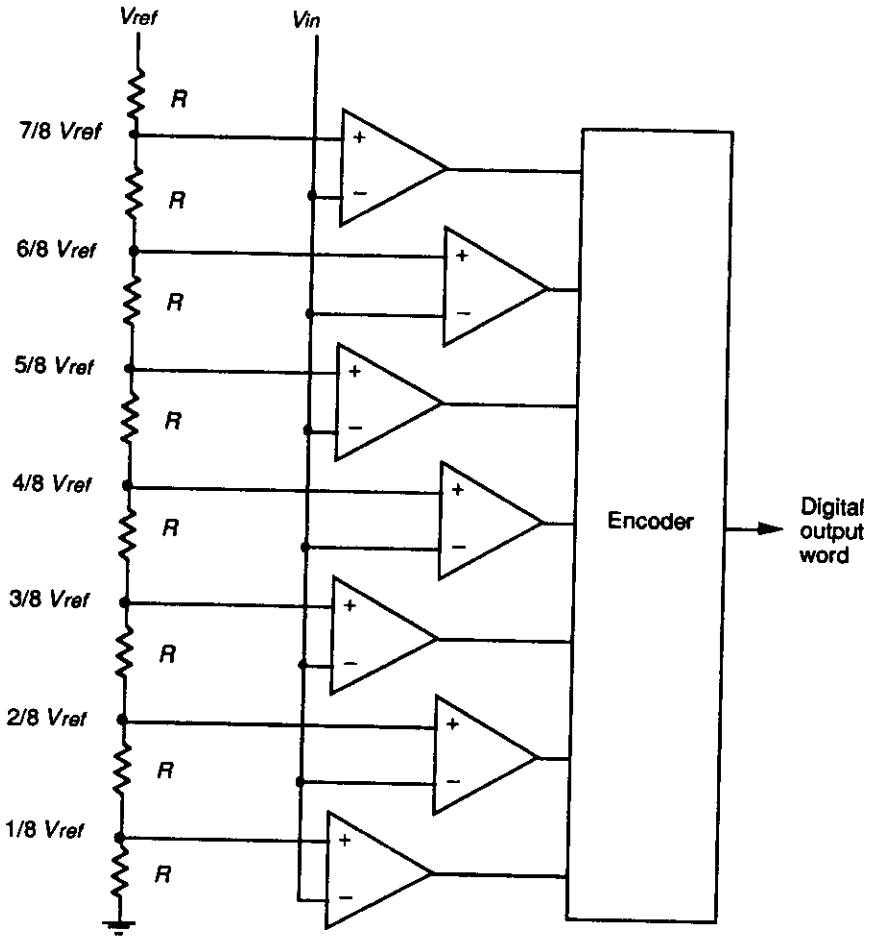


Figure 3.15 A 3-bit flash A/D converter.

3.4.4 Sample-and-hold circuit

Since the conversion from an analog signal to a digital signal takes some finite amount of time, it is advantageous to hold the analog signal at a constant value during this conversion time. Figure 3.16 shows a simple sample-and-hold circuit that can be used to sample the analog signal and freeze its value while the A/D conversion takes place.

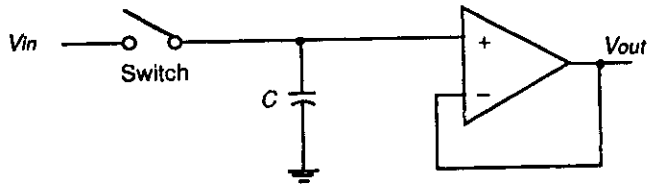


Figure 3.16 A simple implementation of a sample-and-hold circuit.

Errors introduced by the sample-and-hold circuit include offset in the initial voltage storage, amplifier drift, and the slow discharge of the stored voltage. The dynamic properties of the sample-and-hold circuit are important in the overall performance of an A/D converter. The time required to complete one sample determines the minimum conversion time for the A/D. Figure 3.17 illustrates the acquisition time (t_a) and the settling time (t_s).

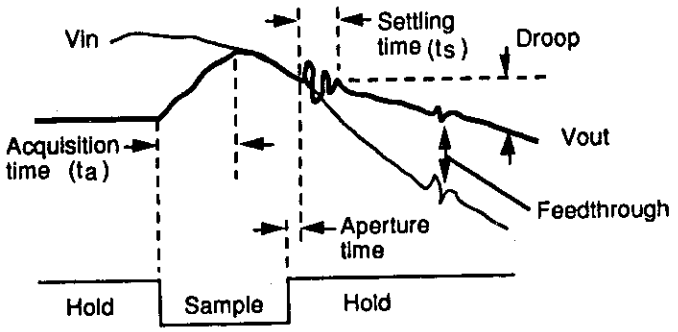


Figure 3.17 Sample-and-hold input and output voltages illustrate specifications.

3.4.5 Analog multiplexer

When several signals need to be converted, it is necessary to either provide an A/D converter for each signal or use an analog multiplexer to direct the various signals to a single converter. For most biomedical signals, the required conversion rates are low enough that multiplexing the signals is the appropriate choice.

Common analog multiplexers utilize either JFET or CMOS transistors. Figure 3.18 shows a simple CMOS analog switch circuit. A number of these switches are connected to a single Vout to make a multiplexer. The switches should operate in a break-before-make fashion to ensure that two input lines are not shorted together. Other attributes to be considered include on-resistance, leakage currents, crosstalk, and settling time.

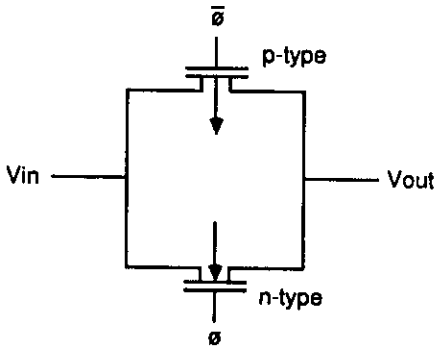


Figure 3.18 A simple CMOS analog switch. The basic functional block of a CMOS analog multiplexer.

3.4.6 Amplifiers

The biomedical signal produced from the sensor or electrode is typically quite small. As a result, the first step before the A/D conversion process is often amplification. Analog amplification circuits can also provide filtering. Analog filters are often used to bandlimit the signal prior to sampling to reduce the sampling rate required to satisfy the sampling theorem and to eliminate noise.

General

For common biomedical signals such as the ECG and the EEG, a simple instrumentation amplifier is used. It provides high input impedance and high CMRR (Common Mode Rejection Ratio). Section 2.4 discusses the instrumentation amplifier and analog filtering in more detail.

Micropower amplifiers

The need for low-power devices in battery-operated, portable and implantable biomedical devices has given rise to a class of CMOS amplifiers known as micropower devices. Micropower amplifiers operate in the weak-inversion region of transistor operation. This operation greatly reduces the power-supply currents required and also allows for operation on very low supply voltages (1.5 V or even lower). Obviously at such low supply voltages the signal swings must be kept small.

3.5 LAB: SIGNAL CONVERSION

This lab demonstrates the effect of the sampling rate on the frequency spectrum of the signal and illustrates the effects of aliasing in the frequency domain.

3.5.1 Using the Sample utility

1. Load the UW DigiScope program using the directions in Appendix D, select **ad(V) Ops**, then **(S)ample**. The Sample menu allows you to read and display a waveform data file, sample the waveform at various rates, display the sampled waveform, recreate a reconstructed version of the original waveform by interpolation, and find the power spectrum of this waveform. The following steps illustrate these functions. You should use this as a tutorial. After completing the tutorial, use the sample functions and go through the lab procedure. Three data files are available for study: a sine wave, sum of sine waves of different frequencies, and a square wave. The program defaults to the single sine wave.

2. The same waveform is displayed on both the upper and lower channels. Since a continuous waveform cannot be displayed, a high sampling rate of 5000 Hz is used. Select **(P)wr Spect** from the menu to find the spectrum of this waveform. Use the **(M)easure** option to determine the frequency of the sine wave.

3. Select **(S)ample**. Type the desired sampling rate in samples/s at the prompt (try **1000**) and hit **RETURN**. The sampled waveform is displayed in the time domain on the bottom channel.

4. Reconstruct the sampled signal using **(R)ecreate** followed by the zero-order hold **(Z)oh** option. This shows the waveform as it would appear if you directly displayed the sampled data using a D/A converter.

5. Select **(P)wr Spect** from the menu to find the spectrum of this waveform. Note that the display runs from 0 to one-half the sampling frequency selected. Use the **(M)easure** option to determine the predominant frequencies in this waveform. What is the difference in the spectrum of the signal before and after sampling and reconstruction?

6. These steps may be repeated for three other waveforms with the **(D)ata Select** option.

3.5.2 Procedure

1. Load the sine wave and measure its period. Sample this wave at a frequency much greater than the Nyquist frequency (e.g., 550 samples per second) and reconstruct the waveform using the zero-order hold command. What do you expect the power spectrum of the sampled wave to look like? Perform **(P)wr Spect** on the sampled data and explain any differences from your expectations. Measure the frequencies at which the peak magnitudes occur.
2. Sample at sampling rates 5–10 percent above, 5–10 percent below, and just at the Nyquist frequency. Describe the appearance of the sampled data and its power spectrum. Measure the frequencies at which peaks in the response occur. Sample the sine wave at the Nyquist frequency several times. What do you notice? Can the signal be perfectly reproduced?
3. Recreate the original data by interpolation using zero-order hold, linear and sinusoidal interpolation. What are the differences between these methods? What do you expect the power spectrum of the recreated data will look like?
4. Repeat the above steps 1–3 using the data of the sum of two sinusoids.
5. Repeat the above steps 1–3 using the square wave data.

3.6 REFERENCES

Allen, P. E. and Holberg, D. R. 1987. *CMOS Analog Circuit Design*. New York: Holt, Rinehart and Winston.

Cromwell, L., Arditti, M., Weibel, F. J., Pfeiffer, E. A., Steele, B., and Labok, J. 1976. *Medical Instrumentation for Health Care*. Englewood Cliffs, NJ: Prentice Hall.

Oppenheim, A. V. and Willsky, A. S. 1983. *Signals and Systems*. Englewood Cliffs, NJ: Prentice Hall.

Tompkins, W. J., and Webster, J. G. (eds.) 1981. *Design of microcomputer-based medical instrumentation*. Englewood Cliffs, NJ: Prentice Hall.

Tompkins, W. J., and Webster, J. G. (eds.) 1988. *Interfacing sensors to the IBM PC*. Englewood Cliffs, NJ: Prentice Hall.

3.7 STUDY QUESTIONS

3.1 What is the purpose of using a low-pass filter prior to sampling an analog signal?

3.2 Draw D/A converter characteristics that illustrate the following errors: (1) offset error of one LSB, (2) integral linearity of ±1.5 LSB, (3) differential linearity of ±1 LSB.

3.3 Design a 4-bit charge scaling D/A converter. For $V_{ref} = 5$ V, what is V_{out} for a digital input of 1010?

3.4 Why is a dual-slope A/D converter considered an accurate method of conversion? What will happen to the output if the integrator drifts

3.5 Show the path a 4-bit successive approximation A/D converter will make to converge given an analog input of 9/16 V_{ref}.
3.6 Discuss reasons why each attribute listed in section 3.3 is important to consider when designing a biomedical conversion system. For example, size would be important if the device was to be portable.
3.7 List the specifications for an A/D conversion system that is to be used for an EEG device.
3.8 Draw a block diagram of a counter-type A/D converter.
3.9 Explain Shannon's sampling theorem. If only two samples per cycle of the highest frequency in a signal is obtained, what sort of interpolation strategy is needed to reconstruct the signal?
3.10 A 100-Hz-bandwidth ECG signal is sampled at a rate of 500 samples per second. (a) Draw the approximate frequency spectrum of the new digital signal obtained after sampling, and label important points on the axes. (b) What is the bandwidth of the new digital signal obtained after sampling this analog signal? Explain.
3.11 In order to minimize aliasing, what sampling rate should be used to sample a 400-Hz triangular wave? Explain.
3.12 A 100-Hz full-wave-rectified sine wave is sampled at 200 samples/s. The samples are used to directly reconstruct the waveform using a digital-to-analog converter. Will the resulting waveform be a good representation of the original signal? Explain.
3.13 An A/D converter has an input signal range of 10 V. What is the minimum signal that it can resolve (in mV) if it is (a) a 10-bit converter, (b) an 11-bit converter?
3.14 A 10-bit analog-to-digital converter can resolve a minimum signal level of 10 mV. What is the approximate full-scale voltage range of this converter (in volts)?
3.15 A 12-bit D/A converter has an output signal range of ±5 V. What is the approximate minimal step size that it produces at its output (in mV)?
3.16 For an analog-to-digital converter with a full-scale input range of +5 V, how many bits are required to assure resolution of 0.5-mV signal levels?
3.17 A normal QRS complex is about 100 ms wide. (a) What is the American Heart Association's (AHA) specified sampling rate for clinical electrocardiography? (b) If you sample an ECG at the AHA standard sampling rate, about how many sampled data points will define a normal QRS complex?
3.18 An ECG with a 1-mV peak-to-peak QRS amplitude is passed through a filter with a very sharp cutoff, 100-Hz passband, and sampled at 200 samples/s. The ECG is immediately reconstructed with a digital-to-analog converter (DAC) followed by a low-pass reconstruction filter. Comparing the DAC output with the original signal, comment on any differences in appearance due to (a) aliasing, (b) the sampling process itself, (c) the peak-to-peak amplitude, and (d) the clinical acceptability of such a signal.
3.19 An ECG with a 1-mV peak-to-peak QRS amplitude and a 100-ms duration is passed through an ideal low-pass filter with a 100-Hz cutoff. The ECG is then sampled at 200 samples/s. Due to a lack of memory, every other data point is thrown away after the sampling process, so that 100 data points per second are stored. The ECG is immediately reconstructed with a digital-to-analog converter followed by a low-pass reconstruction filter. Comparing the reconstruction filter output with the original signal, comment on any differences in appearance due to (a) aliasing, (b) the sampling process itself, (c) the peak-to-peak amplitude, and (d) the clinical acceptability of such a signal.
3.20 An IBM PC signal acquisition board with an 8-bit A/D converter is used to sample an ECG. An ECG amplifier provides a peak-to-peak signal of 1 V centered in the 0-to-5-V input range of the converter. How many bits of the A/D converter are used to represent the signal?
3.21 A commercial 12-bit signal acquisition board with a ±10-V input range is used to sample an ECG. An ECG amplifier provides a peak-to-peak signal of ±1 V. How many discrete amplitude steps are used to represent the ECG signal?
3.22 Explain the relationship between the frequencies present in a signal and the sampling theorem.
3.23 Describe the effects of having a nonideal (a) input filter; (b) output filter.

3.24 What sampling rate and filter characteristics (e.g., cutoff frequency) would you use to sample an ECG signal?
3.25 What type of A/D converter circuit provides the fastest sampling speed?
3.26 What type of A/D converter circuit tends to average out high-frequency noise?
3.27 In an 8-bit successive-approximation A/D converter, what is the initial digital approximation to a signal?
3.28 A 4-bit successive-approximation A/D converter gets a final approximation to a signal of 0110. What approximation did it make just prior to this final result?
3.29 In a D/A converter design, what are the advantages of an R-$2R$ resistor network over a binary-weighted resistor network?
3.30 For an 8-bit successive approximation analog-to-digital converter, what will be the next approximation made by the converter (in hexadecimal) if the approximation of (a) 0x90 to the input signal is found to be too low, (b) 0x80 to the input signal is found to be too high?
3.31 For an 8-bit successive-approximation analog-to-digital converter, what are the possible results of the next approximation step (in hexadecimal) if the approximation at a certain step is a) 0x10, b) 0x20?
3.32 An 8-bit analog-to-digital converter has a clock that drives the internal successive approximation circuitry at 80 kHz. (a) What is the fastest possible sampling rate that could be achieved by this converter? (b) If 0x80 represents a signal level of 1 V, what is the minimum signal that this converter can resolve (in mV)?
3.33 What circuit is used in a signal conversion system to store analog voltage levels? Draw a schematic of such a circuit and explain how it works.
3.34 The internal IBM PC signal acquisition board described in Appendix A is used to sample an ECG. An amplifier amplifies the ECG so that a 1-mV level uses all 12 bits of the converter. What is the smallest ECG amplitude that can be resolved (in μV)?
3.35 An 8-bit successive-approximation analog-to-digital converter is used to sample an ECG. An amplifier amplifies the ECG so that a 1-mV level uses all 8 bits of the converter. What is the smallest ECG amplitude that can be resolved (in μV)?
3.36 The Computers of Wisconsin (COW) A/D converter chip made with CMOS technology includes an 8-bit successive-approximation converter with a 100-μs sampling period. An on-chip analog multiplexer provides for sampling up to 8 channels. (a) With this COW chip, how fast could you sample a single channel (in samples per s)? (b) How fast could you sample each channel if you wanted to use all eight channels? (c) What is the minimal external clock frequency necessary to drive the successive-approximation circuitry for the maximal sampling rate? (d) List two advantages that this chip has over an equivalent one made with TTL technology.

4

Basics of Digital Filtering

Willis J. Tompkins and Pradeep Tagare

In this chapter we introduce the concept of digital filtering and look at the advantages, disadvantages, and differences between analog and digital filters. Digital filters are the discrete domain counterparts of analog filters. Implementation of different types of digital filters is covered in later chapters. There are many good books that expand on the general topic of digital filtering (Antoniou, 1979; Bogner and Constantinides, 1985; Gold and Rader, 1969; Rabiner and Rader, 1972; Stearns, 1975).

4.1 DIGITAL FILTERS

The function of a digital filter is the same as its analog counterpart, but its implementation is very different. Analog filters are implemented using either active or passive electronic circuits, and they operate on continuous waveforms. Digital filters, on the other hand, are implemented using either a digital logic circuit or a computer program and they operate on a sequence of numbers that are obtained by sampling the continuous waveform. The use of digital filters is widespread today because of the easy availability of computers. A computer program can be written to implement almost any kind of digital filter.

There are several advantages of digital filters over analog filters. A digital filter is highly immune to noise because of the way it is implemented (software/digital circuits). Accuracy is dependent only on round-off error, which is directly determined by the number of bits that the designer chooses for representing the variables in the filter. Also it is generally easy and inexpensive to change a filter's operating characteristics (e.g., cutoff frequency). Unlike an analog filter, performance is not a function of factors such as component aging, temperature variation, and power supply voltage. This characteristic is important in medical applications where most of the signals have low frequencies that might be distorted due to the drift in an analog circuit.