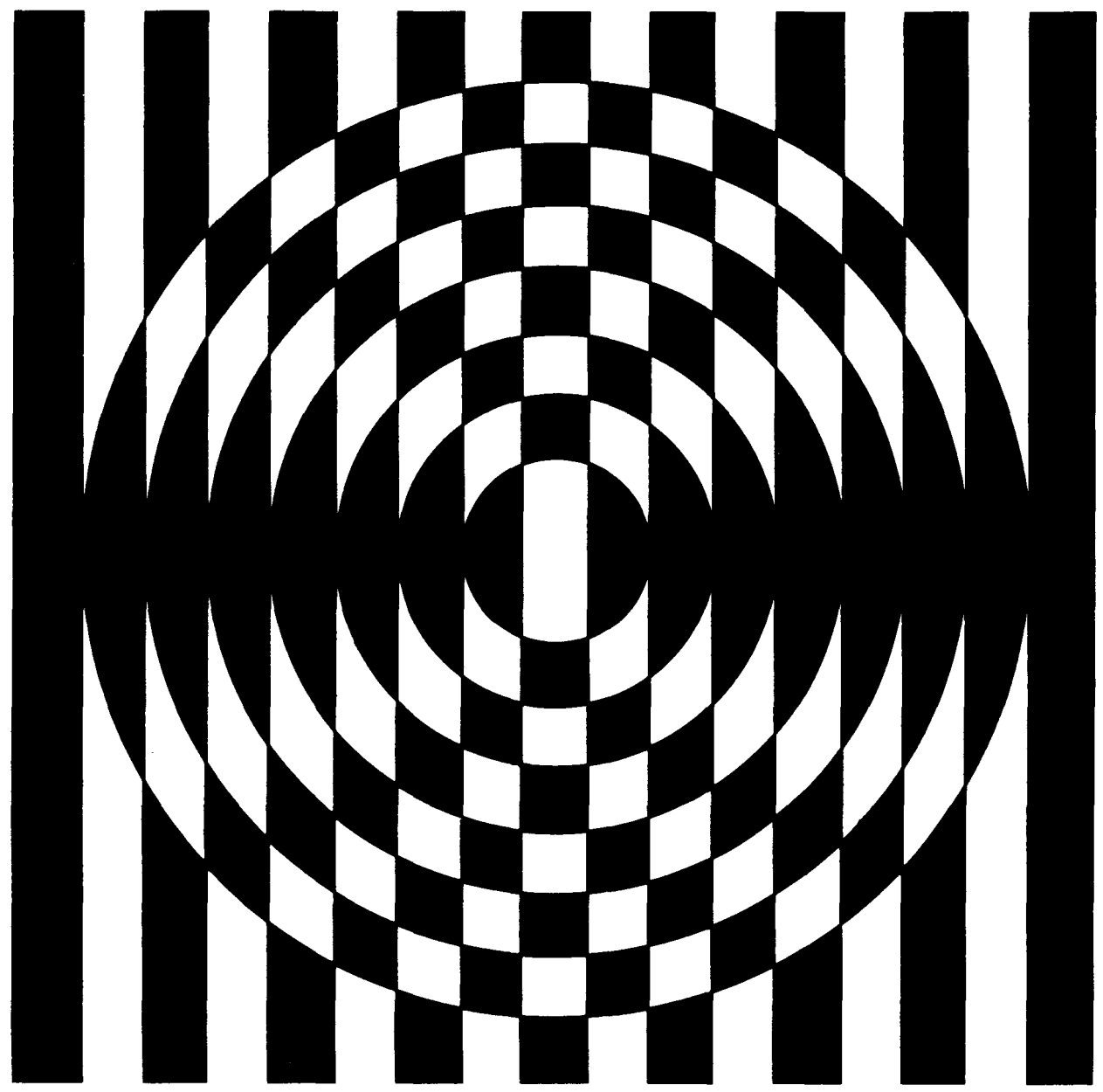


**Minicomputers
in the
Digital Laboratory
Program**

April 1972

Cosine Committee

Commission on Education



MINICOMPUTERS IN THE DIGITAL LABORATORY PROGRAM

An Interim Report of the
COSINE COMMITTEE
of the
COMMISSION ON EDUCATION
NATIONAL ACADEMY OF ENGINEERING
2101 Constitution Ave.
Washington, D.C. 20418

April 1972

Task Force on Minicomputers (VII)

Taylor L. Booth, Chairman, University of Connecticut
C. Gordon Bell, Carnegie-Mellon University
Cecil H. Coker, Bell Telephone Laboratories
Robert M. Glorioso, University of Massachusetts
Edward J. McCluskey, Stanford University
Frederic J. Mowle, Purdue University
David M. Robinson, University of Delaware

COSINE TASK FORCE PUBLICATIONS

Task Force I	Some Specifications for a Computer-Oriented First Course in Electrical Engineering. September, 1968.
Task Force II	An Undergraduate Electrical Engineering Course on Computer Organization. October, 1968.
Task Force III	Some Specifications for an Undergraduate Course in Digital Subsystems. November, 1968.
Task Force IV	An Undergraduate Computer Engineering Option for Electrical Engineering. January, 1970.
Task Force V	Impact of Computers on Electrical Engineering Education—A View from Industry. September, 1969.
Task Force VI	Digital Systems Laboratory Courses and Laboratory Development. March, 1971.
Task Force VII	Minicomputers in the Digital Laboratory Program. April, 1972.
Task Force VIII	An Undergraduate Course on Operating Systems Principles. June, 1971.

These reports have been prepared under the auspices of the Commission on Education of the National Academy of Engineering. Commission policy is to encourage the exploration of new ideas in engineering education. The Commission has been kept informed of the discussions of the COSINE Committee but has taken no position on its reports or recommendations.

The work of the COSINE Committee and these publications are supported in part by the National Science Foundation under Contract NSF-C310, Task Order No. 161

MINICOMPUTERS IN THE DIGITAL LABORATORY PROGRAM

1-0 Prologue

The widespread availability of relatively low cost minicomputers has led to their use as an integral part of many systems. Minicomputers have taken over the function of such special purpose devices as process controllers, communication switching systems and system monitoring units. They have made such tasks as automated typesetting or automatic instrumentation control systems practical.

It is now highly desirable, that undergraduate electrical engineering students be introduced to the minicomputer and learn how it can be used as a system component. To meet this need the undergraduate curriculum must be designed so that a student has an opportunity to work with a minicomputer as an integrated part of his educational program. The availability of one or more minicomputers is an absolute necessity for a computer engineering curriculum.

The introduction of minicomputer concepts into a departmental curriculum requires a considerable amount of effort by the faculty as well as the commitment of financial resources by the department. In March, 1971, a COSINE report, Digital Systems Laboratory Courses and Laboratory Development, [Ref. 2] presented a preliminary discussion of the place of the minicomputer in the undergraduate laboratory program. At that time, it was recommended that a separate COSINE Task Force should consider the problem of bringing minicomputers into the undergraduate program. Therefore, Task Force VII—Minicomputers—was established in the fall of 1971 to study two major areas:

- A. How should the minicomputer be used in the undergraduate classroom and laboratory program?
- B. What type of minicomputer facilities are needed to carry out the recommended program?

The first question is treated in Part I. Two types of applications are considered. A student in the computer engineering area will need a program that allows him to master all aspects of hardware, software and system design. The minicomputer, therefore, provides him with a unique educational tool to explore all of these areas. Other students in electrical engineering may not want or need to go into this depth of knowledge in the use of minicomputers. However, it is important for them to obtain an understanding of the place of minicomputers in general systems as well as the capabilities and limitations inherent in these computers. This section presents the different types of courses and laboratory work required to satisfy both of these needs.

Part II considers the problem of specifying the facilities and equipment needed to implement the various aspects of the recommended program. It provides a general discussion of the factors that should be considered in selecting a minicomputer and indicates the problems that are likely to be encountered in day-to-day operation of the machine. No attempt is made to recommend a specific computer, how-

ever, general system configurations needed to meet a variety of educational goals are recommended.

USE OF MINICOMPUTERS IN UNDERGRADUATE ELECTRICAL AND COMPUTER ENGINEERING COURSES

2-0 Introduction

Previous COSINE task force studies [1,2,3] have considered the types of courses and curricula desirable to introduce computer concepts and computer engineering at the undergraduate level. Underlying all of the recommendations in these reports is the implicit assumption that the student will have access to various types of computer facilities as an integral part of his studies. The form and complexity of these facilities depends upon the particular courses offered and the type of program undertaken by a student.

An examination of the various task force reports indicates that special computer facilities must be provided to service the needs of the following general concept areas:

- A. Computer programming and computing
- B. Machine level programming and computer organization
- C. Systems programming and operating systems
- D. Project work involving the computer as a system element.

Each of these areas requires a computer facility with special features. For this report two classes of students are considered; those planning to enter careers in computer engineering and those interested in the use of computers as a tool to solve problems in other areas. This section discusses the computer facilities needed by students in each of the above areas and indicates the role that the minicomputer can play in satisfying these needs.

2-1 Introductory Programming Courses and Computing

All engineering students require an introductory programming course in which they learn a higher level language, such as FORTRAN, BASIC, PL/I or ALGOL, and how to use a computer to help in the solution of engineering problems. Most introductory programming courses make use of either the batch or time sharing facilities of the local computer center to carry the programming load of these courses since this is often the most economical way to satisfy this need. During recent years a number of minicomputer manufacturers* have developed small, relatively low cost time sharing and batch systems that can be used as the computer facility for the introductory programming course. These minicomputer systems are particularly attractive for small school operation. However, under most conditions the abil-

* Digital Equipment's TSS/8 and PDP-11/45 and HP's 2116 systems are typical of such systems.

ity to use a minicomputer as a computational device and number cruncher should be thought of, at most, as a supplement to the facilities of the Computer Center.

Except in the situation where a minicomputer has been purchased with the particular goal of providing some type of general purpose computation service, its main use should be as a laboratory instrument. This report is addressed primarily to the needs of those courses that use the minicomputer as a laboratory device. Therefore the problem of developing computer facilities to handle the introductory programming courses and departmental computational needs is not considered.

2-2 Machine Language Programming and Computer Organization

The ever-increasing number of minicomputers in industry and government makes it desirable to incorporate at least some minicomputer concepts into the undergraduate engineering curriculum. Modern communications, control, manufacturing and testing systems often rely heavily on a minicomputer as part of the system. The market for minicomputers which are used as an integral part of some system by an Original Equipment Manufacturer (O.E.M.) is now greater than the market for minicomputers used in the "stand-alone" mode. Applications of this type require a deeper understanding of computer organization and programming than is presented in an introductory programming course.

Almost every electrical engineering student will need a basic understanding of both the operational and hardware characteristics of a computer and the concept of assembler language programming. A student planning on a career in computer engineering must go far beyond this basic material and develop an in-depth knowledge of all aspects of computer hardware and software design.

A complete understanding of these ideas cannot be obtained by reading texts and instruction manuals or listening to lectures. A student must have the experience of developing, testing and running a number of complete programs at the machine level that involve the use of all the different machine features including input/output (I/O) devices. These skills can be developed by having the student cover (at least) the following topics in a set of programming experiments. References [4], [5], [6] and [13] discuss these problem areas in greater detail.

1. Basic Operation of a Mini-Computer

One of the chief advantages of a mini-computer is the relative simplicity of its machine language instruction set. The instruction set is usually restricted to data transfer instructions between arithmetic register(s) [accumulator(s)] and memory, addition, a number of test and skip instructions for conditional transfers, instructions for transmitting information over the I/O channels, and a series of micro-instructions which manipulate the contents of the arithmetic register(s), including an instruction to HALT or STOP the execution of a program. For this reason, programming in machine language gives the student an excellent opportunity to learn and appreciate the relatively small set of

basic functions performed by a digital computer (or digital system). Therefore, the first topic to be covered should provide a student with an understanding of a computer from the register level. This can be accomplished by requiring a simple program to be written (such as computing the sum of m-numbers or displaying a changing pattern of lights), then hand-loaded into core, executed, and provided with the necessary test and HALT instructions to stop the computer at the completion of the program.

2. System Software and Assembly Language Programming

The programming of most mini-computers for use in a system is carried out using the assembler language associated with the computer. The writing and debugging of assembler programs is simplified if the student masters the operation of the assembler used by the computer to process assembly language programs as well as such system programs as editors and debugging programs. Thus, after the student has been introduced to the basic operation and instruction set of the computer, he should carry out a series of exercises that will teach him how to develop more extensive programs using the computer's assembly language.

3. Programming Techniques and Data Representations

To use a computer effectively, a student must develop such programming skills as counting, indexing, the use of indirect addressing, the use of pointers, looping, and the construction and use of subroutines. These skills, while important in themselves, also illustrate the trade-offs possible between execution time and the amount of storage required to perform a given function using different algorithms. In addition, the introduction of algorithms raises the question of data representation, symbolic data and numerical data (fixed point, floating point, signed numbers).

The problems associated with the mismatch in timing characteristics between a computer and its peripheral input and output devices must be covered. Solutions to these problems such as the use of flags and buffers (asynchronous operation) should be investigated. Once again the student is faced with the question of data representation, this time converting from the internal code of a computer to the external code used by the peripheral device (such as ASCII for the ASR33 teletype) and vice versa. Techniques for packing and unpacking words when converting from one code to another should be included here.

The classroom and laboratory work necessary to cover these topics should form an integral course (as it should and usually does in an Introductory Programming Course) rather than being presented as two separate courses. For those topics which only involve writing assembler language programs to learn about assemblers or various data structures, it may be expedient to have the students run their programs on a machine which is simulated on the main campus computer. The development of such a simulated program is a straightforward task and several programs of this type are available [SEDCOM Ref. 4, MOHAC Ref. 6, MIX Ref. 7, SAMOS Ref. 12].

For those topics dealing with computer organization and

input/output operations it is vital that the student program and run his own programs directly on a computer. This opportunity for "hands-on" programming experience is an important teaching aid. Experience at a large number of schools has shown that the intimate relationship that develops between the student and the machine is highly motivational. A computer does not allow "partial credit". A stupid mistake, because of carelessness or fuzzy thinking, is forcefully brought home to the student when the computer destroys his program or does not respond in the expected manner. This immediate feedback is invaluable not only because it forces the student to develop proper work habits but also because it quickly indicates those concepts and ideas that he does not fully understand.

A hands-on mode of operation requires that the computer be available on a stand-alone basis since student induced failures cannot be tolerated on a system where a large number of other users are relying on the computer to carry out their computing tasks. This requirement can easily be met by using a minicomputer. Under this condition it will probably be necessary to have a minicomputer that is dedicated to running these student jobs exclusively for at least some portion of each day. Experience indicates that one minicomputer available 24 hours per day can handle a class of 30 to 50 students depending upon the types of facilities available and the magnitude of the programming tasks assigned the students.

The material discussed above is of an introductory nature. The needs of students wishing to follow a computer engineering course of study or those planning work in areas that make extensive use of minicomputers as an integral part of a system will need additional work. The computer engineering student will be interested in further work in both software design and advanced system applications while the student interested in the use of the computer will only be interested in advanced applications.

2-3 Systems Programming and Operating Systems

The contents of an undergraduate course on operating systems principles has been thoroughly discussed in a recent COSINE report [3]. Thus, this discussion will deal only with the experimental work that should be planned to complement this course. Many of the basic concepts of systems programming can be illustrated by developing programs to be run on the campus computer facility. However, for those students wishing to obtain a deep understanding of operating systems rather than just a general knowledge, it is important that a dedicated computer be available. This facility could either be used in connection with the course or for project work for students who have already completed the course.

The primary problems of operating systems are resource management problems. These problems become especially difficult and challenging when the system is designed to allow more than one user or one I/O device to compete for system resources simultaneously. In such a system, there are two difficult resource management problems: the prob-

lem of scheduling resources to competing requests and the problem of protecting the resources allocated to one user from intentional or unintentional damage by another user.

Models of resource management schemes have provided a basis for analysis of the control properties and the performance properties of the schemes modeled. In addition, models are based on fundamental assumptions about the system being modeled. The validity of these assumptions and the results of the analysis of the models can be better understood if they are investigated experimentally with actual systems. Thus the study of operating systems should include an opportunity to carry out both experimental and theoretical studies on an actual operating system.

Operating systems are structured in such a way that additional special software features can be added to carry out specific resource management tasks without interfering with the overall operation of the operating system. When this can be done the facilities of the computer center provide a means to satisfy the laboratory needs for experimental studies of resource management problems. In addition, existing computer centers provide useful facilities for the analysis of models of the dynamic allocation methods of operating systems. These models are sometimes so complex that successful analysis is only possible via numerical methods or simulation methods.

By carrying out projects involving interaction with the operating system, a student faces and overcomes many of the real problems that must be solved when designing and using an operating system. In a laboratory situation the student must not only design and implement a particular experiment but he must also develop methods to measure the experimental system under actual operating conditions. Thus the laboratory facility used must provide ample room for mistakes and failures. Existing computer center facilities are not designed to serve as laboratories. In particular, experiments that might interfere with the general operation of the computer center cannot be tolerated. This means that a separate experimental computer facility must be available for experimental work in systems programming and operating systems.

Until recently, special laboratory facilities for the study of the major problems of operating systems have not been economically feasible. Computer systems with the hardware features allowing memory protection and scheduling of simultaneous processes have been prohibitively expensive. Minicomputers have now been developed to the point where some of the new models have both the hardware features that make them technically feasible and a price that makes them economically feasible to serve as a laboratory instrument for the experimental study of operating systems. The more flexibility that the mini-computer has, such as memory protection, memory mapping, multiple registers and tape and/or disk storage, the greater the range of experiments that can be performed. Features that are not included on the minicomputer can often be simulated or the student can be asked to devise methods to carry out a given task without using a particular feature. Assignments of the latter type are useful in that they impress upon the student the reasons why particular hardware features are desirable as part of a computer's architecture.

2-4 Project Work Involving the Computer as a System Element

The minicomputer, because of its size and cost, is becoming an integral part of many communication, instrumentation and control systems. The computer engineer must be aware of the complete system. He must have a working knowledge of basic machine organization, assembly level programming, hardware interfacing, software dynamics and the basic principles of operating system design for minicomputers. He must also be aware of the human engineering problems so that the system can be used with a minimal amount of effort by the user.

Many electrical engineering students outside of the computer engineering area will be called upon to design, work with, or help incorporate minicomputers into larger systems after they graduate. It is only logical that a department should provide them with an opportunity to become acquainted with the minicomputer as a system element.

This goal can be accomplished in a number of ways. Those students in a computer engineering program can be expected to develop complete system packages as part of their work on projects involving operating systems. However, they should also gain experience in using the computer as a system element without the necessity of developing all of the system hardware and software needed to make the system operational. The students not in computer engineering should understand the basic operations of a minicomputer as described in Section 2-2. They should be provided an opportunity to develop systems that involve a minicomputer as a functional black box which is already properly interfaced to the other system components. The design task then consists of adjusting the parameters of the system to reach a specified operational requirement, or the development of a program in an appropriate higher level language to allow the minicomputer to perform a needed system task.

A minicomputer installation designed for use as a system tool provides a wide range of experimental possibilities. Its flexibility allows a student to investigate problems for which there are no easy analytical solutions. In particular, there are a number of design tasks that require the selection of system parameters based upon a subjective design criterion such as "it sounds funny but the conversation is perfectly understandable" or, "the response is fast enough to keep the operator busy."

The types of experiments that a student can carry out using a minicomputer are determined as much by faculty interest as the availability of equipment. Experience at many schools indicates that whenever there are one or more faculty members with a strong desire to include project work involving minicomputers into a laboratory then the necessary equipment can be obtained for the laboratory. Since the equipment problem is discussed in detail in Section III, the rest of this discussion will be devoted to a brief summary of some typical projects that may be included in a laboratory program devoted to the use of a minicomputer as a system element.

1. *Communication Systems* — Digital communication systems can be studied quite easily using minicomputers.

The influence of sampling rates, amplitude quantization, data compression techniques or encoding techniques upon overall system performance can easily be studied. In a communication system the quality of the received information for a given application is often based upon subjective criteria. The ability to quickly vary one or more system parameters in order to make comparative judgments of system performance is a particularly effective educational tool.

2. *Control Systems* — A wide range of experiments can be built around a control system in which one of the major components is a minicomputer. The relationship between sampling rates, quantization levels, and the time required to carry out a given computational task upon overall system performance can be nicely illustrated in the laboratory. Different types of nonlinear control techniques can also be illustrated and comparisons between different methods of control are easily investigated.

3. *Signal Processing* — The use of digital filtering and Fast Fourier Transform (FFT) techniques to process information on a real-time basis can be studied by using a minicomputer. For example, problems of speech clipping and processing are easily carried out on a minicomputer equipped with A/D and D/A converters.

4. *Instrumentation Systems* — The minicomputer has had a major influence upon the area of real-time instrumentation. Projects directed at developing a complete instrumentation system involve measurement techniques, interface problems, sampling problems and data reduction problems. In addition, decisions must be made concerning the presentation of the measured results in a useful manner.

5. *Automatic Testing* — The minicomputer is an excellent device to control the automatic testing of components, modules and even complete systems. Projects requiring the design of a complete automatic test system for a given task gives the student the opportunity to cover such tasks as test signal generations, measurement of system responses, developing system diagnosis techniques, data reduction and presentation of results.

2-5 Summary

Table 2-1 summarizes the computer related educational objectives that must be accomplished by an electrical engineering or computer engineering program if it is to prepare a student properly for industry or further graduate study. With the exception of the first two objectives, these educational goals cannot be accomplished adequately in the usual computing center environment. The most effective way to teach students about computer operation is to have a minicomputer available for their use. The alternatives of simulating a minicomputer on a large machine or by means of paper computer studies will not do the job except in those areas that involve strictly software concepts that do not require interaction with the outside world or control of the operation of the host computer. There are a number of valid reasons for this situation.

Minor modifications to either hardware or system software in a large computer system can have detrimental effects on user-programs, particularly if they are initiated by inexperienced persons. Directors of central computing facil-

Table 2-1

Summary of Educational Objectives

1. To provide training and programming experience using high level languages.
2. To provide training and programming experience using machine code and assembly level coding.
3. To teach the student the fundamentals of machine organization.
4. To provide the student with an experimental facility to test the ideas and concepts presented in courses on Operating Systems.
5. To provide the students with an experimental facility to test the ideas and concepts learned in courses dealing with interfacing and data processing.
6. To provide the student with an experimental facility to test out ideas and concepts learned in courses on digital process control, digital testing and equipment monitoring.
7. To provide students with an opportunity to understand by hands on experimentation, the relationships and interactions between hardware and software, and the problems presented by real-time programming problems.
8. To provide students with an opportunity to utilize the computer as a system element.

ities will not allow students to interface hardware to the central computer facilities or to modify existing operating systems. To obtain this type of experience students must be able to work with a computer system that is not used to serve the computer needs of the rest of the university. A separate, stand-alone minicomputer is an ideal solution.

The task of writing software for large computers is a frightful experience for a novice. This is due in part to the difficulty in understanding the basic hardware organization of the larger machine with its many layers of control. On the other hand, the basic minicomputer organization can be quickly understood by the average student. Thus, the ability of the student to understand the complete machine, including its memory organization, register transfer operations and I/O structure, quickly gives him the confidence that he can, indeed, write, with a reasonable expenditure of time and effort, complex programs which actually work.

The ability to get at and hang black boxes on minicomputers is also a necessity if digital control and real-time data processing problems are to be considered. The rather transparent architecture of the minicomputer allows an average engineering student to interface instrumentation equipment to the machine, and write an operating system which will control and exercise this equipment as a total system.

The price of a minicomputer installation is often used to argue against the purchase of a minicomputer by a department. This objection is valid if the minicomputer is viewed as just another laboratory instrument. However this viewpoint does not take into account the tasks that a minicomputer can carry out or the different services that it can

provide to the educational program. When the full flexibility of a minicomputer is considered, it will be realized that a minicomputer can be classified as a complete laboratory rather than simply a laboratory instrument. Consequently, the price of developing a minicomputer laboratory is not out of line with the cost of establishing any other type of engineering laboratory.

This section has presented an overview of the types of experiments that should be part of every educational program designed to introduce the use of minicomputers as a system element. No attempt has been made to suggest specific experiments or projects since each school will have different computer facilities and interest. The next part of this report will discuss the problem of creating a minicomputer laboratory facility that can be used to carry out the kind of laboratory work discussed in this part of the report.

MINICOMPUTER LABORATORY FACILITIES

3-0 Introduction

Once a department decides to buy one or more minicomputers for use in its educational program, there are two important factors that should always be considered:

- A. The system should serve the greatest number of students for the least overall dollar investment.
- B. The amount of computer time needed by the student to carry out an assignment should be minimized.

In terms of machine selection, these factors imply that the system has enough features and equipment flexibility to teach the desired concepts; at the same time it is configured so that the equipment and software minimize the non-educational tasks such as inputting programs, printing of listings, or on-line typing and correction of programs.

The task of selecting a minicomputer is formidable because of the number of different machines and peripheral devices that must be considered and evaluated. This part of the report has been prepared to aid in the selection process and will start with a general overview of the hardware and software features that must be considered in specifying a minicomputer system. With this background it is then possible to recommend specific computer configurations to meet different educational needs within given financial constraints.

Throughout the discussion no attempt is made to recommend a particular manufacturer or computer since each installation must be tailored to suit local needs. The final section will discuss miscellaneous topics concerned with the installation and maintenance of an instructional computer laboratory.

3-1 Basic Computer Structure

The computer is a highly structured device. The major components that may be found in a computer system are illustrated in Figure 3-1. Every computer system will have a *primary memory* which holds programs for the processor (usually called the *central processing unit* or CPU) to exe-

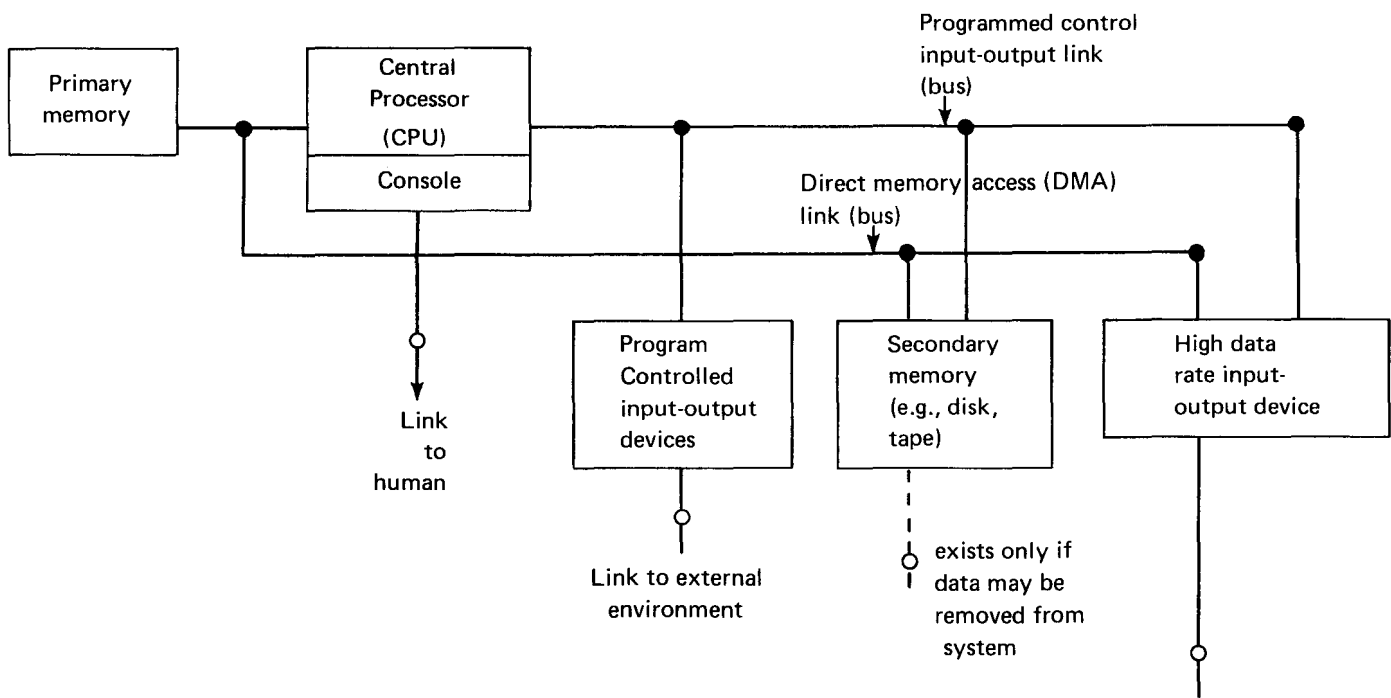


Figure 3-1
Basic Computer Organization

cute. Communication between the computer and the outside world can take place in a number of ways. The processor has a console, consisting of lights and switches, which allow manual communication with the processor. This is a tedious method of information transfer and is used only for such basic operations of starting the computer, making minor modifications to the information stored in the primary memory or indicating a change in operational status of the system. Every computer will have at least one other *input/output* (I/O) device to provide for the major communication between the computer system and the computer user. There are usually at least two distinct modes of I/O information transfer. They are programmed data transfer and direct memory access (DMA) data transfer.

In programmed data transfer the transmission of information between the peripheral devices and the computer is controlled by a set of data transfer instructions issued by the processor. The typical peripheral devices which fall within this category are teletypes, paper tape readers and punches, printers, card reader and punches and analog-to-digital (A/D) converters. Most of these devices have relatively low data transfer rates (that is, 10 to 10,000 characters per second), hence they can be controlled directly by the processor.

The data transfer rates associated with programmed data transfer devices represent one of the limiting factors in an educational computer system. For example a typical computer program of 6,000 characters stored on paper tape requires 10 minutes of read-in time if the standard teletype paper tape reader is used. Special purpose programmed I/O devices are available to speed up the information transfer

rate. These devices are examined in detail in Section 3-4.

The computational work carried out by a computer is mainly accomplished when the processor is operating on information contained in primary memory. If the computing sequence must be temporarily halted while the computer waits for a peripheral device to prepare for an information transfer sequence, the overall computational efficiency of the system drops. Computational efficiency can be improved by using faster I/O data transfer devices. Another way to increase efficiency is to introduce an interrupt capability.

A processor with an interrupt capability carries out its regular computational operation until it receives a signal, called an interrupt, from a peripheral device indicating that the device is ready to execute the data transfer. The processor then interrupts its computational sequence long enough to carry out the required programmed data transfer. It then goes on with the calculation at the point where it was when the interrupt occurred. This type of operation allows the computer to minimize the amount of time that it spends waiting for information transfers to take place.

As higher data rate devices are connected to the computer, there is a need for transmitting data directly to the primary memory without processor intervention. As shown in Figure 3-1 the direct memory access bus is connected directly to the primary memory. In this type of information transfer the processor is unaware of the actual information transfer sequence. Thus the transmission speed is much higher than under programmed data transfer. Some of the typical devices that use this method of information transfer include secondary memory (e.g., disks and tape units) and

very high data-rate links to specialized equipment (e.g., high speed graphic display systems or analog-digital converters).

The above discussion provides a concise overview of the major components of a computer system. In the following sections the important attributes of each component area are discussed and some of the particular attributes that are of importance in developing a laboratory computer facility to serve the educational goals presented in the first part of this report are presented. A much more detailed discussion of the organization of computer systems can be found in references 9, 10 and 11.

3-2 Systems Software

Every instruction that is executed by a computer must be encoded as a sequence of binary digits and stored in the computer's primary memory. Such an encoded instruction is called a *machine language instruction*. The set of all possible encoded instructions form the *machine language* of a computer. All programs to be executed by a computer must ultimately be expressed in the computer's machine language.

Obviously the computer's machine language is not satisfactory for general programming use. To overcome this difficulty every minicomputer manufacturer supplies a set of special programs, called *system software*, that are designed to aid the computer user in developing a program to carry out a specific task.

It is extremely important that the system software available for use by a student to carry out his programming assignments be designed to minimize the amount of time he needs to write and debug his programs. The typical system software available from computer manufacturers is aimed at either the experienced programmer or at users of procedure-oriented languages. For the advanced students this type of system software is adequate. Unfortunately very little attention has been paid to the design of special system software packages to aid in the teaching of introductory software course material. Consequently it will be necessary to supplement the system software supplied by the manufacturer with special purpose system software designed specifically for educational use. This can be accomplished by either developing the needed software "in house" or by locating and borrowing educational software package that have been developed at other schools. The following discussion describes the typical major software programs supplied by manufacturers and gives an evaluation of their usefulness in an educational environment.

3-2.1 ASSEMBLERS

The writing of machine language programs is simplified if short names, called *mnemonics*, are used to identify the basic instructions, memory locations containing specific information and data terms manipulated by the instructions. The set of program statements that can be formed from these mnemonics is called an *assembler language*. A program written in assembler language must be transformed into machine language before it can be executed by the computer. This is accomplished by an assembler program.

An *assembler* takes, as input, a *source program* written in the given assembler language and converts it into an *object program* which can be put into the computer by a loader. The simplest assemblers generate one machine instruction for each source program statement. More sophisticated assembler languages leave many of the routine book-keeping tasks to be taken care of by the assembler program rather than the programmer. The more advanced assemblers allow programmers to assign a name to a sequence of instructions. This sequence is called a *macro*. Whenever the programmer wishes to use this sequence he inserts the macro name into the program. The assembler then inserts the object program instructions that make up the macro into the object program in the appropriate location. Assemblers with this capability are called *macro assemblers*. All manufacturers supply an assembler; some will proclaim weaker or stronger forms of macro capabilities. In truth, macros are not very useful in a small machine. One normally uses subroutines rather than repetitively evoke the same sequence of instructions.

3-2.2 LOADERS

Machine language programs are externally stored on paper tape, punched cards or magnetic tape using a special code which differs from the internal machine language code of the computer. A *loader* program is a program that transfers the externally stored information into the computer's memory.

The loaders supplied by different manufacturers can vary widely in complexity and capability. The simplest are absolute loaders. The absolute location in memory where the stored program is to be placed is included as part of the externally stored information. A relocating loader allows the location of the loaded program to be specified at the time of loading. This allows the use of subroutines from a library of object language programs without having to reassemble each subroutine for every use.

All minicomputers include a basic absolute loader as part of the system software package. Relocating loaders are usually provided only for larger and more advanced systems.

3-2.3 EDITORS

An editor program is a program that allows a computer user to load textual information into a computer and to edit it under control of a typewriter type keyboard. Editors are good for quick program changes on machines that use paper or magnetic tape as the medium of source program input.

An editor program can often represent an unexpected hidden cost to the system. The availability of a good editor program often encourages programmers to put off thinking about program details until they are seated at the computer. Thus a considerable amount of the computer's time may be absorbed in the unproductive task of preparing object programs that could just as easily have been generated off line. Therefore the use of an editor program should be restricted to quick program changes under all situations where the computer has a high use factor.

3-2.4 COMPILERS AND INTERPRETERS

Machine and assembler language programs require a considerable amount of special skill to develop. Several higher level programming languages have been designed so that a program may be written as a set of program statements that closely resemble the form of the mathematical statements that are used to describe the problem under investigation. A special program must then translate the source program into a form that can be used by the computer. There are two general types of translator programs: compilers and interpreters.

A compiler takes the input source program written in the higher level language (usually FORTRAN for most minicomputers) and converts it into an object program that must be put into the computer by a special loader program. Because of the usually severe memory limitations present in minicomputer systems, the compiler is usually of only minimal capability. This usually means that only a stripped down version of the full language can be used. The compiled object program is also often very inefficient since the compiler program is not large enough to carry out any program optimization. Compilers are useful in situations where the need to obtain a working program in a short period of time is more important than developing an optimized program.

An interpreter is a translator program that accepts program statements (usually in BASIC, but possibly FOCAL*), stores them in memory in alphabetic form, and interprets them as numerical commands as it executes them. Interpreters have more flexibility and ease of debugging than compilers do, but they are slower and often do not offer the same range of capabilities as compilers.

One of the biggest problems encountered in the use of a compiler is the time it takes to process a source language program. The time can be quite long depending upon the configuration of the computer used and the design of the compiler. The basic system involving a minimal machine with paper tape input can require up to 20 minutes to load the compiler into memory before processing starts. From this we see that it is impractical to plan on using higher level languages such as FORTRAN on minimal machines. The amount of time can be reduced by using higher speed input devices such as a high speed paper tape reader or a disk.

Interpreter programs also take a considerable amount of time to load on minimal machines. However once the program is loaded it remains in memory while all of the source programs are executed.

An interpreter which allows the insertion of machine language subroutines in the execution sequence is desirable if there is a need to be able to operate any special peripheral devices the computer may have. This is a job for a reasonably skilled programmer.

Many of the interpreters supplied with minicomputers do not have this capability. Like assemblers and compilers, interpreters have a specific range of applications where they are well suited, but for others they are quite useless.

* FOCAL is the interactive language of Digital Equipment Corp.

3-2.5 DEBUGGING PROGRAMS

Debugging programs allow the programmer to run a new program under control to check to see if it runs correctly. A wide range of errors can be corrected by typing simple commands from the keyboard. The use of debug programs is a matter of personal style. They are very useful in introductory courses since they allow the student to investigate the operation of his program under actual operating conditions. Some debug programs work on machine language users programs. Others use the symbol table of the compiler or assembler to allow one to examine quantities by name without looking up the actual memory location in a program listing. In computers with disk or tape bulk storage, the debug packages of some manufacturers eliminate much of the need for printing.

3-2.6 SPECIAL PURPOSE PROGRAMS

A library of math subroutines can be useful even if a compiler language is not the principal mode of operation of the machine. Trigonometric, statistical and polynomial-algebraic subroutines may be useful for projects involving scientific applications. A subroutine library editor is useful for maintenance of the system library in scientific applications. In projects, where programs tend to become large, a good library editor can save machine time by encouraging programmers to prepare private libraries. For other applications, a library editor is less important.

When a much larger computer system is purchased with some form of secondary storage then additional software is needed to make efficient use of the system. A file system or disk operating system (DOS) provides a way of allocating space in a disk and accessing it by name rather than numerical address. A system executive program or monitor provides a way of calling in programs from magnetic tape or disk. A real-time executive provides a means to swap different parts of a large program into and out of main memory from disk — thus allowing secondary storage to substitute for primary memory. "Real time" for one application can be intolerably slow for another. If an application needs job-swapping software, the manufacturer's version of it may still not suit those needs or the machine configuration.

3-2.7 EDUCATIONAL SOFTWARE

The major problem with software supplied by a manufacturer is that each program is designed to carry out one type of task on a large problem. Most educational applications, particularly at the introductory level, deal with short, student generated programs in which it is common to find a large number of errors. Special teaching software is desirable to minimize the amount of time a student must spend in locating and correcting these errors and in testing the program to prove that it is operating properly. Combined editor-assembler-debugging programs are useful for small programs. In this case the student can create, debug and run a machine level program in an interactive manner. Programs of this type are usually not supplied by the manufacturer, except for the larger machines with some form of bulk storage such as a disk, and must be developed locally.

Sometimes useful educational software can be obtained from the user groups that have been formed by the different manufacturers. There is, however, a definite need to develop better ways to handle the exchange of software and software ideas between schools.

The teaching process is another way in which special purpose software can be developed. In each class there are usually a few very enthusiastic students who really enjoy software development projects. If a list of "needed software" is developed, it serves as a source of project work for these students. Whenever any software is developed it is imperative that it be properly documented. This documentation should include a complete commented listing of the program, a flow chart, a discussion of how the program is organized and how it is used. A project should not be considered complete until this documentation is finished. If the software developed has any general use it should be submitted to an appropriate users group for general distribution. The "honor" of having a program selected for submission is often an added incentive for the student to do a good job on the project.

3-3 PRIMARY MEMORY AND CENTRAL PROCESSOR ATTRIBUTES

The primary memory and central processor form the nucleus of any computer system. Once a decision is made concerning these two units a complete pattern is set for the specification of all other system components. This section explores the important attributes associated with primary memory and the central processor and indicates which attributes are of particular importance in selecting a computer for use in an educational program.

3-3.1 PRIMARY MEMORY

The basic information storage element in a computer can store a signal with a value of either 0 or 1. Usually more information than this must be stored. To do this the storage elements are grouped together to form a register. A register with n storage elements is called an n-bit register. The main memory of a computer has a large number of identical registers arranged in an orderly manner, something like a collec-

tion of post office boxes. Most of the memory systems are designed so that each register in the memory is directly accessible to the central processor. Each of the registers in the memory holds one word of information. The word length of the memory is the size of the basic register and the memory cycle time is the time required to access any memory word.

There are three important attributes of primary memory which influence the operation of the computer and the types of problems it can solve. They are:

- a) The word length - measured in bits
- b) The memory size - measured in words
- c) The speed or data rate - measured in terms of cycle time

Currently most primary memories are constructed using magnetic core technology. In the future there will probably be a shift to integrated circuit memories as integrated circuit technology develops better memory units at lower prices.

For every application there can be an optimum word length. However, if the computer is to be used in a wide range of different situations it is not possible to select an optimum length. In this case the criterion is to select the largest word length possible within the financial constraints imposed by the fact that memory price increases directly with word length.

Since word length has a strong influence upon the types of tasks that computers can perform, it has often been used to provide a rough method of machine classification. One classification [Ref. 8] that has been suggested is given in Table 3-1.

The length of a computer word influences how data and the machine language instructions executed by the processor are stored in memory. A word should be large enough to hold and allow manipulation of data (which use multiple words to form larger data strings) and also hold the information needed by the processor to execute an instruction.

Memory size is somewhat similar to word length in that larger memories allow larger programs and more data to be stored. The minimum memory size selected for a general purpose system is 4K words (here K stands for $1024=2^{10}$). Such a size is both large enough for most student projects

Table 3-1

Possible Classification of Small Computers

Name	Word length	Max. addressable primary memory (words)	Number of registers in central processor (words)	Data-type
micro	8 ~ 12 bits	4K	2	integers, words, boolean (bits)
mini	12 ~ 16 bits	32K	2 ~ 4	vectors (via index registers)
midi	16 ~ 24 bits	65K ~ 128K	4 ~ 16	double length integers (and occasionally floating point)

Note: $K = 1024=2^{10}$ when referring to information storage capability.

(since even the most facile programmer can probably code less than 100 instructions per day) and the systems programs necessary to support this programming effort. More advanced applications will require larger memories to hold data and large programs. Currently the cost of add-on memory installed after the initial delivery of the computer is minimal for most new computers. Thus a safe policy may be to start with the minimal memory size of 4K and then plan on adding additional memory as system needs grow.

In selecting a computer system particular attention should be paid to the cost involved in adding extra core memory to the system. In some computers the cost of adding an extra 4K of core memory is almost equal to the price of the basic machine. In some of the newer systems additional core memory can easily be added at moderate cost without any modifications to the original machine.

Memory size is perhaps most important as it affects the availability and capability of software supplied by the manufacturer. Therefore, if one or more manufacturer supplied software packages are considered to be of importance to the overall laboratory effort, it is important to make sure that adequate memory capacity, as well as other required components, is available to accommodate these software packages.

Memory speed, or inversely, the memory cycle time, is a measure of how fast the processor can place information in memory or remove it from memory. This is probably the computers' least important parameter for most of the educational activities considered in this report. Currently the average memory has a speed of about 1 megahertz, or a cycle time of 1 microsecond. Since most of the input-output devices are usually quite slow and only minimal length computational tasks will be performed by the system, it is hard to justify additional cost for a factor of two or even three in speed. There are a few conditions, such as realtime experiments, where high data rates may form a particularly important part of the computational load of the system. For these cases speed might be a much more critical parameter.

3-3.2 THE PROCESSOR

Perhaps the single most important consideration of the central processor is its console. The console is the component through which the user communicates with the computer in the "hands on" mode of operation. The console has switches for information inputs, and lights to view results. It is desirable to have all the registers which may contain meaningful information for the user to be indicated on the console. The switches should be very durable (a common failing of minicomputers) and similarly the lights should be mounted to allow easy replacement. The design should be conservative so the lights will not to burn out readily.

The processor is somewhat less important than the primary memory because it is constrained by memory attributes. The processor word length and speed are almost always locked to the properties of the primary memory associated with the processor. Each processor will have a number of design "features" that are singled out by the

manufacturer as making their particular machine better than that of a competitor. The features of competing processors are often very hard to evaluate. The best (and almost only) method of evaluating processors is to write and run a set of representative programs (called benchmark programs) on each processor and compare overall results. This is extremely difficult to do since these benchmarks must be representative of *all* the types of jobs a machine will run over its lifetime. An even more important benchmark consideration is deciding on the language of the benchmark. For example, while a machine may have a number of nice features, the FORTRAN compiler supplied by the manufacturer may not use them — hence they are worthless if programs being run are written in FORTRAN. The selection of a processor for instructional use can usually be accomplished without this elaborate evaluation process. The main consideration in this case is that the processor have as many features as possible, consistent with other requirements, that will expose the student to a wide range of programming techniques.

The machine language instructions executed by the processor are represented by a bit sequence. These instructions are stored in memory and each instruction will occupy one or more computer words. Thus the memory required (hence, cost) to hold a program to carry out a particular task can vary by a factor of two or three. Therefore the organization of the processor is an important factor in this regard and cannot be totally ignored.

In order to get higher program "code" density (i.e., the number of bits to hold a program that performs a given function), more sophisticated means of accessing memory are usually added to the processor. These features can cloud the initial exposure for the student. Ideally, if multiple computers are available, a student might start with a very simple, minimal computer, such as a PDP-8, and then later be introduced to a computer with more advanced features, such as index registers and multiple general purpose registers. The features relating to program "code" density, are usually features of the instruction set, and these are given in Table 3-2.

The ability of a processor to access and operate rapidly on a large number of data types, such as bytes, character strings, integers and floating-point numbers, is a main feature of a processor. All of the operations needed to operate on any data type can be realized by programs. In many cases special hardware capabilities can be included in the processor to handle special tasks. For example built-in multiply and divide operations are often useful for several types of high-data-rate real-time processing. Similarly, if a machine is to be used for very general computation, floating point arithmetic is useful. The floating point feature is usually not found in current minimal machines, and although important for certain applications, must be carefully justified if an additional cost is involved.

Most basic machines are designed so that any information contained in a memory location can be modified if the instruction being executed by the processor indicates that this should be done. There are many situations where it is desirable to protect the information in one or more memory locations from being changed. This is particularly true

Table 3.2
Computer Features Useful
(and Necessary) to Illustrate
Various Techniques

1. Multiple processor registers. General registers which allow use as accumulators, indexing registers, program base, and stack registers. (Used in larger compilers).
2. Data-types and data operators.
 - a. Stacks. Useful for subroutines calling and storage, and teaching recursion.
 - b. Bytes and character strings. Useful in teaching about translators, and communication equipment.
 - c. Multiply and divide operations. (Problem dictated).
 - d. Double and multiple precision arithmetic.
 - e. Floating point arithmetic.
3. Addressing modes (e.g., immediate, indexed, indirect). Used in larger computers.
4. Program protection and relocation. Various schemes ranging from: a protect bit per word (s), to base and size (relocation), to a set of page registers, to segment name registers are useful. Each allows various (increasingly more complex) principles to be illustrated.
5. Program interrupts. Multiple levels are useful both to illustrate the principle of concurrent operation and for particular applications.

if work is being done on systems that will allow multiple, independent programs to be operated in quasi parallel (i.e., by rapidly switching from program to program). To allow this ability requires that the processor have some form of memory protection capability.

A number of degrees of protection can be provided ranging from fully programmed protection (which may require a significant amount of the processor's capability) to very general hardware, which provides an environment for each program as though it were running by itself. While having this facility generally available may be an asset, it can probably only be justified if the computer is to be used for serious multiprogramming operating system projects.

Some newer computers are organized about microprogramming principles. A small, fast, internal (microprogrammed) processor is used to provide a basic set of hardware operations. Each instruction in the instruction set that the machine user sees is implemented as a microprogram consisting of a sequence of these basic hardware operations. The definition of the user instruction set is accomplished by defining a microprogram for each instruction. These programs are stored in a fast, usually read-only, memory. The user's program is then written in terms of the instructions contained in the resulting instruction set. This program is held in a conventional memory and is executed by the processor in the conventional manner.

Microprogramming can be used to provide variations in the instruction set available to the user. However, in most current computers the lower level microprograms reside in read-only memory, thus it is usually a difficult and/or expensive task to change them. From the viewpoint of pro-

gramming there is no conceptual difference between a microprogram machine and a conventional machine. In fact the instruction set of many minicomputers are often very close to the basic instruction set of a microprogrammed processor. Should a microprogrammable machine be selected where the user can change the instruction set, then one must be prepared to provide local software support for the machine.

If the microprogramming of the instruction set can be easily changed this becomes a very handy feature for experimental work by students interested in computer design. However a standard instruction set should be defined for all other applications.

One of the major tasks of the processor is to be able to communicate with external I/O devices. These devices may be the standard I/O devices needed for regular operation of the computer or they may be specially designed devices developed by students as part of a project. Thus the ease with which a processor can be interfaced to these devices is an important feature of the processor and should be carefully considered.

Normally data is transferred by (1) the processor issuing a command to the input-output device, (2) the device activating an interrupt sequence that causes the processor to issue the proper information transfer commands to the input-output device, (3) the processor and the device operating independently and in parallel. In all these cases the flexibility of the computer is strongly influenced by the ease with which these different types of transfers are performed.

The first two transfers are programmed information transfers. The processor must be able to select the device to be used and then send out the necessary commands to accomplish the transfer. A processor should be selected which is designed so that a wide range of devices can easily be interfaced to the computer and controlled by the programmed information transfer facility.

The interrupt facility simply switches the program currently operating to an alternative program which is associated with the interrupting device. The important attribute is the interrupt response time, which is the time between interrupt request and the time the interrupt is acknowledged. Since all computers have this facility, it is hard to differentiate among them. However, both the capability for the interrupt to transfer program control to a unique program associated with the interrupting device, and the provision of a number of interrupt levels so that relative priorities between input devices may be established are desirable features. They simplify coding, illustrate good programming principles and require less programming overhead than simpler interrupt methods. For the neophyte, however, programming for a multilevel interrupt system can be quite difficult and take considerable debugging time to correct programming errors. All of the features of an interrupt can be realized on a one level interrupt system by programming techniques at an increase in interrupt response time.

For the parallel transfer of information it is necessary to have a direct memory access link (bus) associated with the processor. This link is required if high data rate devices such as disks or tape units are to be used with the com-

puter. The capability of using direct memory transfer is also desirable for use in more advanced laboratory work.

The physical characteristics of the direct memory access link, like that of the program controlled input-output link, are the most important consideration. In some computers both links use the same cables or bus (e.g., a unibus). This allows the student flexibility in designing the interface without drastically affecting the cost. In other computers the direct memory access link is completely separate and requires special interface units for each device that is to be connected to the link.

An often overlooked property of any interfacing arrangement is the way that the physical connections are made between the computer and the external device. This feature is a very important part of an input-output interface that will be used by students in carrying out projects. Here, the user should consider the cables, connectors, logic type, and general construction of the machine as it provides for ease of interfacing. The cables and connectors must be robust if they are to stand up to normal student use and misuse.

Student projects will take on a variety of forms. The interface should be designed so that student designed and

other special hardware can be easily attached to the computer. In particular, a well defined standard interfacing policy should be established so that the students will have a design standard to follow when they are developing a system that must be interfaced to the machine.

Table 3-3 summarizes the important attributes of the primary memory and central processor that should be considered. The relative importance of each attribute in the evaluation of a computer is indicated on a scale of 1 through 4. A rating of 1 indicates that the attribute is very important to the overall operation of the system and should be considered in detail when evaluating a given computer system. A rating of 2 indicates that the attribute should be carefully considered while an attribute with a 3 rating may be of importance under some conditions. An attribute with a rating of 4 is usually of minimal importance in selecting a computer system.

3-4 Basic Input/Output Devices

The most critical part of a computer system, from an educational viewpoint, is the method used to provide communication between the student and the computer. Each

Table 3-3
Computer Attributes and Their General
Importance to Teaching

Component	Technology	Attributes	Rating
Primary memory	Core and possibly integrated circuitry	Memory size: 4K*, 8K, >16K words	1
		Word length: 8,12,16*,18,24 bits	2
		Memory speed: 1.0 microseconds	4
		Console (light and switch robustness)	1
Central Processor (CPU)	Integrated circuitry	Instruction-set (general Data-types: words, integers, real addressing modes (methods) registers (accumulators, index, general purpose)	2
		Protection and relocation	3
		Microprogramming	4
		Programmed input-output link	1
Input/output access link		Interrupt response time	3
		Cables, connectors, and access	1
		General	3
Direct memory access link		Cables, connectors and access	1
		Maximum speed	3

* Average value

Rating Scale

1 - Very important; 2 - Look at carefully; 3 - May be important; 4 - Usually ignore.

computer system will need some equipment to accomplish the following basic input/output tasks (roughly in the order of importance)

- A. A means to load system programs such as assemblers and editors.
- B. A means to read student prepared source programs.
- C. A means for the student to interact with the computer.
- D. A means for printing or displaying listing of complete programs.

There are a number of ways that these tasks can be performed. The selection of input/output devices for a particular system should therefore be approached from the viewpoint of long range educational efficiency rather than the minimization of initial cost.

Every computer system must provide at least the following minimal input/output capability.

- A. A keyboard which may be used to enter information and interact with the computer.
- B. A printer or other alphanumeric display device to output messages to the user.
- C. A means for entering machine readable programs.
- D. A means for outputting permanent machine readable programs.

The standard input/output device usually found on a minimal minicomputer system is a teletype with a paper-tape reader-punch. Such a unit carries out all of the necessary input/output tasks. A teletype is, however, quite slow and leads to inefficient use of the system for the type of tasks considered in this report.

The three basic methods of external program storage that are in widespread use on minicomputers are:

- A. Paper tape
- B. Cards
- C. Magnetic tape (small tapes and cassettes)

The selection of a method of external program storage not only influences the type of input/output device that must be included on the computer but also influences the type of equipment that is needed for the off line preparation of programs.

This section will present a discussion of the advantages and disadvantages of a number of different input/output devices. The emphasis of the discussion will be upon the usefulness of each device as part of an educational computer system. The comments on overall performance capabilities of the different devices are independent of system application. However, the usefulness of a particular device as a system component is strongly influenced by planned system use. Thus, if the planned computer system must serve other needs, every attempt should be made to match the I/O equipment to the type of information that must be handled by the system. In selecting any I/O equipment for student use the ruggedness of the equipment is perhaps one of the single most important characteristic to be considered.

3-4.1 TELETYPES

Because of their low initial cost, teletype units have become standard as the input/output device of many computer systems. The unit actually consists of two subunits; an input unit consisting of a typewriter type keyboard and paper tape reader, and an output unit consisting of a printer and a paper tape punch. One input and one output unit can usually operate in parallel at a rate of 10 characters (or bytes of 8-bits) per second.

The keyboard/printer is used to allow the user to communicate directly with the computer while the paper-tape is the basic media for external program storage.

The ten character/second typing rate is not a problem when information is to be typed into the computer since this far exceeds the typing capability of any student. This rate can be painfully slow when a student is waiting for the computer to read in a paper tape, to print out information or to punch out a paper tape. In fact this low data rate can introduce serious inefficiencies in the use of the computer systems.

A considerable amount of programming effort involves the use of system programs such as assemblers, editors or compilers. A typical program of this type may contain 6,000 characters. The task of reading in this program requires 10 minutes of read-in time at a 10 character/second reading rate. The FORTRAN compiler for some computers requires almost a half hour of read-in time each time a FORTRAN program is processed. Similar delays occur when a large program listing must be printed or a long object program must be punched on paper tape.

If a teletype must be used as the primary input/output device, then every effort should be made to minimize the amount of time needed to execute an information transfer operation. This can be accomplished by eliminating as many unnecessary characters as possible from the information stream (i.e., comments from listings, long character strings for mnemonics, spaces on tapes etc.).

There are a number of ways the bottleneck introduced by the teletype can be removed. Unfortunately an increase in speed also means an increase in price. Several useful alternatives to one or more input/output functions of the teletype will now be considered.

3-4.2 HIGH SPEED PAPER TAPE READERS AND PUNCHES

The standard input on most minicomputers has been paper tape. One way to speed up the input process is to use a high-speed photo-electric paper tape reader. These readers are 30 times faster than the standard teletype reader. Thus a program that requires 30 minutes to be read on a teletype reader can be read in 1 minute by a high speed reader.

High-speed paper tape punches can speed up the output process of generating a paper tape. Unfortunately high-speed punches are only about 5 times faster than a teletype punch. If a choice must be made it is much more desirable to add a high speed reader to the system before a high-speed punch since much more information must be read from paper tapes than punched onto paper tapes in carrying out normal assignments.

When paper-tape is used in a system one or more spare teletypes should be available for the offline preparation of source tapes and printing of output listings. The off-line generation of tapes has the drawback that it is very hard to make corrections on tapes.

3-4.3 CARD READERS AND CARD PUNCHES

There are two types of card inputs that are available: mark/sense cards and punched cards. Mark/sense cards require the student to use a pencil to darken boxes on the card corresponding to the character or character sequence he wishes to indicate. This method of card preparation is very tedious and time consuming. Thus, mark/sense cards are of limited value in any application involving programs with a larger number of statements. The addition of a mark/sense card reader should only be considered if there is a special need for such a device to handle a specific class of problems not associated with the general education activities planned for the system.

There are now several low cost card readers that are available for use with minicomputers. These readers have about the same data rates as a high speed paper tape reader. The main problem with a punched-card card-reader is the need for some method to prepare the cards.

An offline card punch is more expensive than a teletype. Cards are much more convenient for correcting and editing but listing and proofreading cards offline is a problem. The editing speed advantage may allow one keypunch to serve as well as two teletypes.

The card reader should be selected as the high speed input device if it is possible to *easily* use the facilities of the computer center to take some of the load off the minicomputer. This can be done in a number of ways. For example, the minicomputer's assembler program can be simulated on the computer in the computer center. This program can then be used to carry out preliminary assembly and debugging (through removal of errors in the assembly language statements) tasks. This simulation program will probably have to be developed locally. It is also possible to use the computer center's printer to obtain offline listings of programs. This approach will reduce but not eliminate the need for paper tape or magnetic tape input/output facilities on the laboratory computer.

Whenever a card reader is used there is always a need to have a local card punch facility for students to prepare cards. This may be either a standard key punch (which is rather expensive) or one of the manual punches (which requires the student to punch the actual binary code) that are available. The addition of a card punch to the minicomputer as an output device is very expensive.

3-4.4 CASSETTE AND OTHER SMALL MAGNETIC TAPE UNITS

In recent years the cassette magnetic tape has been adapted for use as a data storage medium. A digital cassette tape unit is about as expensive as a card reader and has comparable data rates. Cassette tapes provide a very compact means for storage of large amounts of program material. In the past some of these tape units have displayed

marginal reliability. Thus particular care should be taken if it is decided to purchase this type of unit. The technology in this area is advancing rapidly and some of the early problems will be eliminated in the near future if they have not already been corrected.

Cassette tapes are ideal for the storage of computer generated information and system programs since information is easily transferred between the computer and a tape. At present the preparation of cassette tapes offline or the offline listing of a program stored on a tape presents a problem. However, if offline preparation and listing techniques become better developed this form of input and output will be one of the better all around input/output mediums for minicomputers.

Other conventional tape units are also available for use on minicomputers. Some of these units use IBM compatible tapes and provide both high data transfer rates and high reliability. The cost of these tape units is somewhat higher than a cassette unit but the greater flexibility, speed and usefulness in some particular applications may justify the added cost.

3-4.5 DISPLAY TERMINALS AND HIGH SPEED PRINTERS

When a student is waiting for output information from the computer, the 10 character/second printing rate of the teletype can be painfully slow. There are two types of outputs needed. During the time a student is working with the computer there is a need for the transfer of messages between the computer and the student informing him of the operation of his program. These messages are for information purposes only and do not need to be saved. At the end of a program development or the completion of a given task there is usually a need for a complete program listing or the printing of an extensive amount of other information. This information must usually be in a permanent form.

A number of graphic display terminals using storage CRT displays are currently on the market for a little more than twice the price of a teletype. These terminals are well suited to the normal computer-student communication requirements. The use of these terminals can often reduce the need for a large number of print-outs during a work session. At the end of the session a single print-out will be needed.

A teletype is a very slow device for printing complete listings of programs. If a teletype must be used then every effort should be made to reduce the amount of printed output information that is required in a program. If some form of high speed output is available, such as a high speed tape punch or card punch or a cassette tape unit, it may be possible to record the desired output information and do the actual printing offline. The facilities at the computer center might help with this problem.

Several moderate cost, medium-speed printers have recently been introduced. These printers, which operate at speeds up to 300 lines a minute, can significantly reduce the amount of time the computer must spend generating printed output. If a number of minicomputers are used in the same laboratory, the overall cost can be reduced if the

printer can be used on a time-shared basis by all of the computers.

3-5 Expanded Memory and Secondary Storage

A minimal computer system with 4K words of memory is satisfactory for most introductory and many intermediate and advanced projects. Extra memory, however, introduces many advantages and should be considered next in importance to having high-speed input/output capabilities.

Memory capacity can be expanded in two ways: by the addition of more core memory or by the addition of secondary memory units such as magnetic disks, magnetic tapes or magnetic drums. Three important parameters—cost, speed and capacity—must be considered in selecting the method to be used to add additional memory capacity to the system.

3-5.1 CORE MEMORY

Additional core memory will be needed as the system expands and more sophisticated projects are undertaken. When secondary memory units are added, extra memory space is required to hold the software systems associated with these units if adequate memory space is to be provided for users' programs. At least 8K of memory should be present on an expanded system. However, 12K or 16K of memory would be even better. The cost of expanding memory beyond 16K is usually not justified for an educational system and the money is better spent on expanding the secondary storage capabilities of the system or adding other features.

The addition of core memory is the most costly way of expanding the system's overall storage capability. Thus one approach might be to first expand the memory to 8K, then add some form of secondary storage and then add an additional 4K to 8K of core memory.

3-5.2 SECONDARY MEMORY

The availability of secondary memory in a computer system provides considerable flexibility for the system. Secondary memories may provide one or more of the following four functions:

- (a) Holding programs and program libraries for fast, easy access.
- (b) Bulk storage of data for later use.
- (c) As a media to temporarily hold parts of one or more programs while another program is being run.
- (d) Communication with another computer (i.e., a reel of magnetic tape can be used to transfer information from one computer to another).

All of these functions provide for greater flexibility and speed when the computer system is in use.

There are three common types of secondary memory. They are:

- (a) magnetic tape
- (b) magnetic disks
- (c) magnetic drums

As has been previously discussed, magnetic tape units can serve as large information storage units. Another advan-

tage is that the tapes can be easily transported from one unit to another or interchanged. In this way each user can build up an individual file on a tape and then take it with him when he completes his use of the computer.

Disks and drums have essentially the same capability as far as the system is concerned. They can store a large amount of information in a relatively small space. The storage medium (disk or drum) is permanently connected to the computer and hard, if not impossible, to remove.

The access time is the most critical parameter of secondary memory. This is the time between a request for information and the availability of the information. Disks and drums have the lowest access time since all information is no more than a revolution away. The access time of a tape is a variable depending upon how much of the tape must be searched before the information is located.

Data in secondary memory is usually stored in blocks. Once a block of information is found the characters that make up the information stored in a block must be transferred to the main memory. The data transfer rate (in characters/second) then indicates how fast information can be transferred once the block in which it is stored is located. The data-rate characteristic of a device is of lesser importance because it is usually dominated by access time.

The total time, t , to transfer a block of information is therefore

$$t = \text{access time} + \frac{\text{block-length (in characters)}}{\text{data-rate (in characters/second)}}$$

Table 3-4 compares the relative capabilities of the different secondary memory units. These units may serve any one of the following four functions:

- p - long term program storage
- d - data (result) memory
- s - swapping memory which holds programs that are in the process of being executed but are currently inactive
- i - interchange memory which is used to carry data from one machine to another

For most teaching use, program memories are important because they facilitate handling of the programs themselves. Memories used for holding data are also important because these can extend the computer's primary memory by providing a temporary storage for later processing, (usually magnetic tape). The use of memories for interchanging information may be important since this permits using larger facilities. For example, small computers have traditionally controlled and gathered data at an experiment; the data is then carried to a large machine for final analysis. Disk memories are often used to hold one or more programs which are temporarily inactive. This use probably occurs in the larger computers' configurations when several users are simultaneously sharing a single computer (time-sharing).

3-6 Other Peripheral Devices

The usefulness of a minicomputer as a teaching device depends to a considerable extent upon its flexibility. There are a number of special purpose peripheral devices that allow the minicomputer to be connected as part of a larger

Table 3-4
Attributes of Secondary Memory

Device	Memory Size (char)	Access Time	Data - rate	Function**
Cassette Tape		Sequential	~300ch/sec	p
Addressable* Mag. Tape	500K	Sequential at read speed	15K ch/sec	p,d
Conventional* Tape	1-10 million	Sequential at read speed	4K - 1.2 million ch/sec	p,d,i
Disk pak*	800K up	1/4 sec or faster	24K - 120K ch/sec	p,d,s
High per- formance disk, or drum	32K-800K	1/20-1/60 sec	120K - 3 million ch/sec	d,s

* Usually requires DMA channel because of high data rates.

** p - long term program, d-data (result) memory, s-swapping memory which holds programs that are in the process of being executed but which are currently inactive, i - interchange memory which is used to carry data from one machine to another.

external system. The range and number of these peripherals determines, to a large extent, the types of problems that can be solved with the computer.

The price of many of the peripheral devices are decreasing while overall performance is increasing with time. Therefore it is often advantageous to wait until a particular device is needed before it is purchased. This, of course, assumes that the extra installation cost associated with adding a device at a later time will not make the addition of the device excessively more than the cost of the device if it is installed at the time the original machine is purchased.

The following devices have been found to be particularly useful. The operational characteristics included in the discussion represent those that can be expected from current technology.

A. Analog-to-digital (A/D) Converters: An A/D converter transforms an analog signal into a digital signal. The important parameters are sampling rate (samples/second), conversion time (seconds) and resolution (bits/sample). A 10 bit A/D converter with a conversion time of 10 to 20 microseconds should be sufficient for most applications.

The flexibility of an A/D converter can be improved by using a multiplex switch to allow more than one input analog signal to be connected in sequence to the A/D converter. A 4 to 6 channel switch is usually sufficient for most applications.

B. Digital-to-analog (D/A) converters: A D/A con-

verter transforms a digital signal into an analog signal. The important parameter is response time which should be in the range of 3 to 5 microseconds.

It is often desirable to have from 4 to 6 D/A output channels. Each channel should be connected to a separate buffered D/A converter so that the output voltage remains constant until the computer changes the value of the output by placing a new digital signal in the buffer.

C. External Timer or Clock: An external timer or clock is almost a mandatory option because it provides the computer and program with knowledge of actual time. Experiments in communication, instrumentation, control, etc., usually require "timing" capabilities.

To provide maximum flexibility the clock should be under computer control so that it can be started or stopped on command. It should also be able to generate timing pulses strings with a controllable period from a few microseconds to seconds.

D. Relay Closures and Contact Sensors: These devices provide switching that can be used to start and stop external processes or sense when a certain task has been completed.

E. Graphical Input Devices: A number of special purpose tablets have been developed to allow a user to enter graphical information concerning the position of a special "pen". These devices are of somewhat limited use on small computers but they can be very useful for

situations where a large amount of graphical input information must be entered into the machine.

F. Displays: CRT displays provide a very useful output medium when used in connection with D/A converters. For small computers the storage scope display units are best since they do not need to be continually refreshed.

G. Plotters: X-Y plotters are useful when graphical data must be recorded. Some of the X-Y plotters designed to work with analog signals are able to work directly with signals from D/A converters. However others are not suited to this type of signal. Any plotter used with a computer should have a pen that can be raised and lowered under computer control.

H. Student Interface Connection: When a student is working on different types of input/output projects, it is necessary to allow him to use all of the different input/output facilities. Thus, standard connectors should be available so that the student can interconnect the equipment that he has developed to the appropriate data or control line in the computer. *All control and data lines brought out to this connector should be fully buffered and protected so that the student cannot damage the computer through carelessness or mistakes.*

It is also useful to have one or more data buffer registers that can be used as general interface registers. In this way it is possible to minimize the amount of hardware that the student has to develop and allows him to concentrate upon the particular information transfer task on which he is working.

All of these devices can usually be added to a computer system after the initial purchase of the machine. Thus, in selecting a basic system the cost of future expansion to include one or more of these devices should be considered.

3-7 Additional Considerations in Minicomputer Selection

The task of selecting a minicomputer involves a number of considerations. The primary objective is to purchase the best computer to satisfy a given educational objective. However, there are a number of other related factors that will determine the effectiveness of the computer in satisfying the educational objectives for which it was purchased. This section will consider these factors.

3-7.1 SITE SELECTION

Although the physical size of most basic minicomputers is such that they could easily be placed on a bench in a laboratory, past experience at a number of schools has indicated that it is best to buy a minicomputer installed in a regular floor-rack. This approach serves a number of important purposes.

The computer is the focal point of a number of activities. If it is rack mounted, an area in the laboratory can be reserved exclusively for its use. Since there are usually a large number of people using the computer, an area should be selected that will provide enough work space for working on programs while not actually on the machine. It is also desirable to have the area organized in such a way that it is possible to use the computer "after hours". Many depart-

ments have found that their minicomputer is used 24 hours a day if provisions can be made to allow students access to the machine at night.

One of the main advantages of a minicomputer is its usefulness as a systems element. As indicated previously, a number of projects will probably be developed where the computer must be interfaced to other hardware. If the computer is rack mounted it will take up less than a fourth of the rack area, leaving enough space for added peripherals as well as a standard interface arrangement that will make it easy to interface the computer to student projects.

The amount of floor area needed for the minicomputer depends to a considerable extent upon its use. If the main use is as a stand-alone computer for running software projects then adequate space must be provided for easy access to the input terminal and any peripheral devices which the operator must use.

When the minicomputer is interfaced with external hardware it is necessary to have a considerable amount of space available in the laboratory to accommodate the hardware. There are two ways to meet this need. If the laboratory is large enough, laboratory work bench area in the lab can be assigned to each project. The whole project is then developed in the laboratory and can be connected to the computer through a cable run from the work bench to the computer. One should not, however, underestimate the amount of space needed for each project. Experience has shown that each project will typically require a six foot long work bench area. This space will be tied up by this single project until it is completed. When the hardware development cannot be accommodated in the computer room, provision should be made for the use of either a rolling table to bring equipment into the room or cable connections between the hardware area and the computer area.

The interaction of the computer with experiments in remote laboratories is often necessary when the experiments involve special purpose equipment. For low speed transmission up to 1200 baud, telephone lines and data modems can be used. Twisted pair cable can be used up to approximately 1 MHz. Coax cable can be used for data up to several megahertz.

Conventional room air conditioning can normally handle the temperature/humidity requirements for basic minicomputers. As the system increases in size, a more extensive air conditioning system will be necessary. The air conditioning system must be designed so that it is operational during the winter as well as the summer.

All computers are sensitive to dust. Thus it is desirable to use floor tiles to cover a concrete floor. It is not necessary, however, to have special raised floors typical of those found in computer centers.

Standard laboratory electrical service is adequate for most minicomputer installations. However if the system is expanded to include a number of peripheral devices, such as disks or display systems, it is often desirable to provide electrical service that is totally dedicated to the use of the computer system.

Other than the simple precautions listed above, a minicomputer installation can be placed almost anywhere that is convenient. Experience has shown, however, that the

computer area will experience a high use factor and prior planning should include the consideration of general convenience of the system users and the possibility for easy future expansion.

3-7.2 DAY-TO-DAY HARDWARE MAINTENANCE

When a minicomputer becomes the central instructional component in an educational program, it is imperative that the amount of down time due to computer failure and routine servicing be minimized. This means that some form of maintenance program must be developed consistent with available financial resources and educational needs.

Experience has shown that once a minicomputer has run for a few months and the minor system adjustments and modifications made, the basic central processor will run virtually error free. The majority of unscheduled down time will be caused by mechanical failures in peripheral components, teletypes, readers, punch tapes, etc. In addition, there are several preventative maintenance tasks that must be performed periodically on the system.

The most desirable situation is to be able to employ a regular technician to maintain the complete computer installation. As a minimum, the technician should have a general electronic background with, hopefully, some experience in digital systems. The necessary training to maintain the specific computer purchased can be obtained by sending the technician to take the standard maintenance course offered by most manufacturers. One limitation of these courses is that they often do not cover the diagnostic procedures and maintenance problems associated with the computer's peripheral components. Maintenance courses on these devices are usually available at a reasonable cost.

If a technician is not available to assume responsibility for the machine, it is imperative that some faculty member take the overall responsibility. This is not to imply that he should do the actual maintenance but he should have the authority and resources to see that it is done. There are a number of ways that this can be accomplished.

Complete field maintenance contracts are available from most minicomputer manufacturers. These are run on a profit basis and, in general, barring a catastrophic system failure are somewhat expensive. There are, however, some off-setting features if the service center is located reasonably nearby. A complete stock of spare parts does not have to be maintained by the department and the complete field service operation is available to handle unusual problems.

If local conditions do not allow for a full time technician or a full service contract, a graduate assistant assigned full time to the system or a part-time technician should be available. This arrangement should be backed up with the availability of a per-call field maintenance agreement.

When you elect to maintain the system yourself it is advisable to purchase a small quantity of spare parts for the system. (Most field service technicians are more than willing to provide a list of what is needed.) Failed parts not stock-piled can usually be obtained quickly on an exchange basis plus handling charge from a regional service center.

As a precautionary measure, it is often advisable to have a periodic (once a year) preventative maintenance test and

recertification test run on the computer by trained maintenance people.

If graduate students are to be used as system engineers, the problems of graduation and documentation will have to be faced. The former problem can be alleviated by using more than one student employed on an overlapping basis. Maintaining adequate documentation on all system modifications (hardware and software) is a very difficult problem and can only be solved by using a strong hand. Poor hardware or software documentation can prove to be very costly since a considerable amount of time can be consumed just trying to reconstruct what the system is trying to do if no documentation is available. The problem of poor documentation often occurs if the modification was developed as part of a project laboratory course. Under this situation the satisfactory completion of the course should require complete and acceptable documentation.

If the minicomputer is to be used in a project lab environment with external black boxes hung on the computer, there is the possibility of accidental damage to the computer due to faulty external circuit design. To minimize the possibility of such failures, it is advisable to build a buffer unit between the computer and some sort of universal I/O patch panel or connector. In many cases, failure to adequately protect the computer, will void maintenance contracts.

Lastly, be sure to provide a small budget for the everyday needs of the computer, such as paper, cards, paper tapes, magnetic tapes, etc., which are required to keep the facility running.

3-7.3 UTILIZATION OF THE COMPUTER CENTER

The specialized service of the central computer center can be used in several ways to increase the efficiency of a remote minicomputer facility. However, a great deal of advanced planning and discussion of goals with the computer center staff will probably be required to insure the close working relationship necessary for smooth interaction with the center.

When the minicomputer facilities are not sufficient to handle the introductory machine level programming work it is often possible to use the computer center facilities to relieve some of this load. This can be done by writing an assembler and a simulator for an existing minicomputer which can be run on the center's computer in either a batch or remote time sharing mode. Most of the initial programming experience needed to gain familiarity with machine and assembler language concepts can be obtained in this manner. When the students start using the department's minicomputer, the availability of a minicomputer assembler on the center's computer which yields loadable object code can greatly decrease the time lost loading text editors and assemblers on the minicomputer, thereby allowing for better utilization of the machine.

The use of an alternate assembling procedure is particularly important when large numbers of students are executing programs on the minicomputer. By utilizing the facilities and services provided by a central computer center, much of the text editing and syntactic debugging of programs can be

done offline (i.e., not on the minicomputer).

For large classes of students (30 or more), the initial programming experience should be in a batch processing environment with fast turn around time. The students can then be introduced to hands-on experience when they have demonstrated their ability to write and assemble correct code. If large groups of students are competing for time on a minicomputer, offline programming and editing is a must if the students are to use their on-line time efficiently.

If a variety of simulator programs for existing or paper computers are available on the center's computer, a richer programming experience can be provided for the students. The task of writing either the simulator or assembler for a minicomputer is a good project for students interested in system programming.

Another way of increasing the capability of a minicomputer is to connect the computer to the central computer by means of a time sharing or remote batch link if such services are available. To be effective, the link should be reasonably fast, 9,600 baud or better. By means of this connection, the minicomputer can make use of the peripheral equipment (card readers, line printers, plotters and bulk storage devices) of the main computer. Also by means of this link, back up number crunching can be provided for real time problems using the minicomputer as a data handling and communications processor.

3-8 Configurations For Specific Applications

The previous discussion presents a comprehensive overview of the various items that must be considered in specifying a minicomputer system for departmental use. The final specification of a particular computer configuration depends upon the educational tasks it must perform, the funds available and local faculty preferences. This section provides a general outline of the type of system configuration that is needed to satisfy a number of different needs.

The range of uses for a departmental computer can be represented by the following five typical tasks (in order of increasing difficulty):

- A. *Introduction to machine level computing.*
Experience in hands-on operation of a small machine. Machine language and assembler language programming. Use of standard input-output techniques.
 - B. *Study of computer hardware and interfacing.*
Techniques of interfacing different devices to the computer. Study of hardware-software interaction and trade-offs.
 - C. *Study of the computer as a system element.*
Emphasis is upon the use of a computer as a component part of real-time systems use in instrumentation, control and signal processing.
 - D. *General project laboratory work.*
The computer becomes a tool that can be used as part of a system to solve problems.
 - E. *Study of advance computing techniques.*
Investigating of such techniques as data structures, file management, multiple-task processing, job scheduling and organization of operating systems.
- In most situations a single computer will have to be

available to carry out more than one of the above tasks. The following discussion indicates the particular minimal capabilities necessary for each task and suggests the desirable features to add to a system as greater capability is desired. Experience has shown that many schools start out with a basic system and then slowly expand their systems as time and funds permit. Thus, in designing a system, possible future growth should be considered along with current needs.

3-8.1 INTRODUCTORY LABORATORY CONFIGURATIONS

Introductory instruction requires only a small computer. The main consideration is to minimize the amount of time the student spends carrying out the routine tasks of processing programs and obtaining listings. This means that any of the standard minicomputers with 4K memory are satisfactory. A machine with a common or typical instruction set is preferable to a very unique machine, but this is a secondary consideration. Experience in programming one machine readily generalizes to others.

The minimal input/output needs can be served by using the teletype that is almost standard with these machines. As indicated previously there are a number of relatively inexpensive additions that can be added to obtain greater I/O efficiency. Either a high speed paper tape reader or card reader should be the first improvement considered. Normally the paper tape reader is the best selection unless it is possible to use the facilities easily of the computer center to help with the preparation and processing of punched cards. A high speed paper tape punch or card punch is desirable but not as important as the high speed reader.

Some type of graphic display terminal is also a desirable addition to the minimum system if any growth is anticipated. This terminal can, however, be postponed at some sacrifice to efficiency of system use. Another way to improve efficiency of system use is to add secondary storage in the form of an addressable magnetic tape device. This could either be a cassette tape unit or a low cost standard computer tape unit.

3-8.2 INTERMEDIATE LABORATORY CONFIGURATIONS

The structural features needed for a study of computer hardware and interfacing techniques are the same as for the introductory programming work. The major exception is that the machine should be chosen with a view toward being representative and versatile in methods of hardware interfacing.

Preferably the machine should allow interfacing at a basic level such as an "I/O bus", rather than only through special device interface cards or "Channels". The machine should be capable of information transmission through some form of "direct memory access" block transfer mechanism, as well as a programmed word-by-word mechanism. Standard connectors should be included that make all control and data lines available for student projects. The computer should also have an input/output interrupt facility.

Priority interrupt capabilities are useful but not absolutely necessary at this level.

It is also useful to have one or more data buffer registers that can be used as general interface registers. In this way it is possible to minimize the amount of hardware that the student has to develop and allows him to concentrate upon the particular information transfer task on which he is working.

3-8.3 THE COMPUTER SYSTEM LABORATORY

The two previous computer configurations center around the computer almost as an end in itself. When the computer is considered from the viewpoint of being a system element, then the emphasis centers on what the computer can do rather than upon trying to acquaint the student with specific details of machine level programming and debugging. Laboratory work involving the computer as a system element should spare the student from as many programming details as possible so that there will be time to accomplish a higher-level goal.

A measure of the effectiveness of a laboratory assignment is the time it takes to go from the start of a task to the end result. Thus it is desirable to allow the student to use a higher level language to carry out as many programming tasks as possible.

This approach reduces the number of human errors that must be located and corrected approximately in proportion to the number of program statements necessary for a given job. The speed of making these corrections will depend upon the computer and the language.

An interpretive language such as BASIC or FOCAL is very useful in those experiments where processing time is not important. Interpretive language allows the fastest turn-around time from making a correction to seeing the result. It also provides useful debugging assistance as well. Unfortunately an interpreter gains this capability at the expense of execution speed. For those problems where an interpretive language is too slow, a compiler language such as FORTRAN or ALGOL is useful. The speed of developing programs can be improved by having predeveloped assembly language subroutines available to drive problem oriented peripheral devices connected to the system.

Efficient interpreters and compilers require at least 8K to 12K of primary memory. In this case the larger the memory, the greater the range of information processing tasks that can be undertaken.

In addition to the need for expanded primary memory there is an increased need for some form of secondary memory such as a standard tape unit or a disk. This addition is desirable since many interesting system applications involve the handling of a large amount of real-time data. The availability of secondary storage also makes it easier for students to use the system software packages in developing programs. For example the time needed for a source-level correction in a FORTRAN or ALGOL program can be accomplished in a few minutes if a disk or tape resident compiler is available. The same correction can take considerably longer if a paper tape or punched card dependent software system is used.

Signal processing experiments force minimum requirements on word length. Because of roundoff errors, 16 bits is really minimal for integer arithmetic programming of digital filters, if signal-to-noise ratios of 40 or 50 DB. are to be maintained. A machine with 12 bit words will thus require the use of double precision arithmetic to achieve desired results. Additional equipment for the survey machine should include such things as relays and sense lines, A/D and D/A converters, voltage controllable oscillators, digital A-C voltmeters, real-time clocks, displays and other equipment for computer-controlled tests.

3-8.4 PROJECT LABORATORIES

Equipping a projects laboratory can be more expensive than equipping an applications survey laboratory. The emphasis on elimination of programming details is much the same as in the survey course, but the requirements of projects may be greater and more varied. Experiments of the applications survey laboratory can be tailored to the capabilities of the machine. The projects laboratory machine, however, is subject to the outside requirements of real problems that must be unique in their own right.

Projects are apt to need at least one of the following:

1. Data Storage—either high-capacity magnetic tape or disk.
2. Data Transportation to the Computer Center—either magnetic tape or a data link.
3. Large Memory—there is no upper limit here, but a reasonable choice is 12 or 16 K with a fast disk and procedures for overlaying different phases of the program.
4. High Speed—an interpretive language is entirely too slow for some signal processing.
5. Longer Word Length—16 bits can be inadequate for many signal-processing needs. Differences of large numbers occur, for example, in high-rejection filters and in filters with low-frequency singularities, compared to the sampling rate.

A 16K machine with disk, magnetic tape and hardware floating point or integer double precision arithmetic is usually too rich for a one time budget. Thus a reasonable approach is to start out with a small but expandable machine that is useful for projects of immediate interest, and hope to expand as new needs and funds develop.

Software for the projects machine has much the same character as an applications survey machine. There will probably be more need for math library subroutines and higher computation speeds. If the machine has a disk so that the turn-around time for a source-level correction can be kept low, the best mode of operation is to use a compiler language like FORTRAN augmented with assembly-language subroutines where speed is needed. An editor for preparing and modifying subroutine libraries will be useful—both for the system library and for individual student private libraries.

3-8.5 ADVANCED PROGRAMMING LABORATORY

A course in Advanced Programming Techniques will require a machine of intermediate size—about 8K. File man-

**Table 3-5
Desirable Attributes of A Computer for a
Given Task**

Teaching Task	Rating 1-Necessary 2-Strongly desirable 3-Good to have	Component	Attribute
Introductory	1	Primary memory	4K, 12 or 16-bit any speed
		CPU	Preferably simple and straight-forward
		I/O Peripherals	Teletype, paper-tape reader and punch, or card reader
	3	Secondary memory	Addressable tape or other
Hardware and Interfacing	1	Primary memory, CPU, I/O Devices	Same as above, PLUS:
		I/O methods	An assortment of low-level I/O methods; 1 or more interrupts
	3	Secondary memory	Addressable tape or other
Computer as a Systems element	1	Memory	12K or more, 16 bits or more. Speed somewhat important
		CPU	Straight-forward, but possibly more complex
		I/O peripheral Special I/O	Cards or fast tape; scope A/D, D/A converters, relays, sense lines; some engineering devices to control
	2	Secondary memory	Tape or pack disk
	3		High-performance disk
Advanced Techniques	1	CPU	8K or more, 16 bit or more. Advanced, complex machine desirable
		I/O	Several external devices of various speeds, capable of interrupting
		Secondary memory	Preferably high performance disk, but pack disk, even addressable tape
	2	CPU	Memory protection and relocation

agement, job scheduling a swap-out experiments will require a disk (or perhaps addressable magnetic tape). Input/output both through direct memory access and word-by-word programmed transfer with interrupts is desirable.

There must be external equipment to be controlled, and equipment to generate and use data at high and low speeds, both continuously and sporadically. This need not be expensive, however; microphones, loudspeakers, lights, photocells, even electric trains serve admirably for this purpose.

When a department reaches the point where it needs a computer facility to study advanced computing techniques it will, of necessity, have one or more faculty members with expertise in this area. These faculty members will be able to advise the department in the development of computer facilities for this advanced level work.

The different possible computer configurations that should be considered for each type of task are summarized in Table 3-5. The relative importance of different components to the teaching task of a given area is indicated by a rating of 1 through 3. A rating of 1 indicates that the component is of central importance, a rating of 2 indicates that it is strongly desirable to have the component, and a rating of 3 indicates that the component is useful to have, but not at the expense of other needs.

IV. CONCLUSIONS

Today's electrical engineer must be familiar with the characteristics and operations of minicomputers. This means that every electrical engineering department should have at least one minicomputer available for use as part of the undergraduate laboratory program. Cost is no longer a limiting factor since it is possible to obtain a basic system for approximately six thousand dollars. Once a basic system is installed it can be slowly expanded by adding peripherals as the use of the system grows.

The development of a minicomputer laboratory facility should be carried out with the same amount of care and advanced planning that goes into the development of any laboratory. The planning of the initial system should be done so that it is flexible enough to allow for future growth. Under ideal conditions this planning would include identification of the faculty members who will be responsible for the development of the system and a long range commitment of funds to insure proper growth and maintenance.

Minicomputers are playing an increasingly important role in many different areas throughout institutions of higher education. It is not uncommon to find minicomputers used in research laboratories to replace special purpose equipment. This means that there is usually a need for a course to introduce students from other fields to the operation and use of minicomputers and how to interface minicomputers to different pieces of equipment. The development of courses to satisfy this need is a valuable way of developing the interdisciplinary interaction that provides interesting projects for student project laboratory work. The good will generated by the successful completion of an interdisciplinary project should not be overlooked.

The increasing attention being paid to minicomputers by both computer manufacturers and peripheral manufacturers means that there should be major new developments that will lower the price and increase the flexibility of minicomputer systems. As these developments occur this will mean that much more extensive problems can be carried out by undergraduate students. It is therefore becoming more important that each electrical engineering department should have at least one faculty member whose interest lies in the computer area to supervise the development of both the minicomputer facilities and the academic programs that use these facilities.

The development of a minicomputer laboratory facility requires a considerable amount of time and effort of one or more interested faculty members as well as a realistic commitment of funds by the department. Laboratory work of this type requires a considerably larger amount of time, effort and money than the development of a lecture course. The time and effort can be reduced if an effort is made to communicate new developments and ideas among the faculty members at the different schools who are working on laboratory developments of this type. This development effort could be helpful in the following ways:

- a) When new educational software is developed for a given computer system the program should be submitted to an appropriate users' group so that it can be made available to others.
- b) A book is needed that discusses the use of a minicomputer as a system element. It should concentrate upon the practical problems that are encountered in applying a computer to realistic real-time system problems rather than a theoretical discussion of computer organization.
- c) Meetings should be organized for faculty members to exchange ideas and experiences about laboratory work. These meetings could be held in conjunction with regional ASEE, IEEE or ACM meetings.
- d) Interesting projects, and the support required to purchase the necessary supplies to carry them out, can often be obtained by working with local industries, public institutions or even other departments in the school. The full potential of this source of projects can only be realized by careful planning and cooperation among all parties involved. The proper selection of the students to work on these projects is critical. An improper match between student capabilities and expected results will leave all parties to the project unhappy.

References

1. COSINE Task Force Report IV—An Undergraduate Computer Engineering Option for Electrical Engineering, January 1970.
2. COSINE Task Force Report VI—Digital Systems Laboratory Courses and Laboratory Development, March 1971.
3. COSINE Task Force Report VIII—An Undergraduate Course on Operating Systems Principles, June 1971.

4. Booth, T. L., *Digital Networks and Computer Systems*, John Wiley, New York, 1971.
5. Gear, C. W., *Computer Organization and Programming*, McGraw Hill, New York, 1969.
6. Reilly, E. D., Jr. and Federighi, F. D., *The Elements of Digital Computer Programming*, Holden-Day, San Francisco, 1968.
7. Knuth, D. E., *The Art of Computer Programming*, Vol. 1, Addison-Wesley, Reading, Massachusetts, 1968.
8. Bell, G., Cady, R., McFarland, H., Delagi, B., O'Laughlen, J., Noonan, R. and Wulf, W., *A New Architecture for Mini-computers—The DEC PDP-11*, AFIPS, SJCC 1970, Vol. 36, pp. 657-675.
9. Bell, G. and Newell, A., *Computer Structures: Readings and Examples*, McGraw-Hill, New York, 1971.
10. Gschwind, H. W., *Design of Digital Computers*, Springer-Verlag, New York, 1967.
11. Foster, C., *Computer Architecture*, Van Nostrand Reinhold Company, New York, 1970.
12. Forsythe, A. I., Keenan, T. A., Organick, E. I. and Stenberg, W., *Computer Science: A First Course*, John Wiley, New York, 1969.
13. Stone H.A., *Introduction to Computer Organization and Data Structures*, McGraw-Hill, New York, 1972.