Computer science in electrical engineering

The computer revolution has made a significant impact on electrical engineering curricula. The electrical engineering department's new role and the need for greater course flexibility in the computer environment are examined

COSINE Committee of the Commission on Engineering Education

The vital role that electrical engineering departments must play in providing undergraduates with special competence in computer sciences is explored. Three related problem areas are discussed: (1) meeting the needs of students majoring in computer sciences in electrical engineering; (2) balancing the treatment of continuous and discrete systems so that students have a background in discrete systems comparable to that which they now acquire in continuous systems; and (3) realizing wider and more effective use of the digital computer as a tool for analysis and design in all engineering courses. Specific suggestions for meeting these needs are offered.

The rapid growth in the accessibility and power of digital computers for purposes of numerical computation, data processing, and retrieval of information is having a deep, though not necessarily uniform, impact on all branches of science and engineering. Although most branches of science and engineering are concerned primarily with the *use* of digital computers, electrical engineering, by virtue of its long standing and deep involvement in information-processing technology, has vital concern not only with the use but, more important, with the conception, design, and construction of digital computers. Moreover, electrical engineering is deeply involved in a wide gamut of areas that border on or are contributory to computer technology, such as integrated circuits, switching theory, and finite-state, control, communication, and adaptive systems.

During the past several years, the rapid growth in the use of computers in science, engineering, and many other fields has tended to shift the emphasis in computer technology from circuit and component design to system organization and programming or, in roughly equivalent but more succinct terms, from hardware to software. This trend has given an impetus to the crystallization of what is now widely referred to as computer sciences—that is, an aggregation of subject areas centering on the use of computers as large-scale information-processing systems.

Clearly, it would be unreasonable to equate the computer sciences with electrical engineering, or to regard it as a subset of the latter. Nevertheless the close relation between the two is presenting electrical engineering departments with a special responsibility for the training of large numbers of computer engineers and scientists. This responsibility derives not only from the close connections between electrical engineering and computer technology, but also from the traditional emphasis in electrical engineering curricula on physical sciences pertinent to information processing; the extensive experience in teaching mathematically oriented subjects relating to signals and systems; the tradition of analyzing system behavior on an abstract level without regard to the physical identity of its variables; the existing expertise in subjects that fall into computer sciences or are closely related thereto; and the vast resources in facilities and faculty which the electrical engineering departments have at their disposal.

The emergence of computer sciences as a highly important field of study, coupled with the growing shift in emphasis in information-processing technology from the analog and the continuous to the digital and the discrete, is creating an urgent need for a major reorganization of electrical engineering curricula. Such a reorganization must, in the first place, accommodate the needs of students who wish to major in computer sciences within electrical engineering. Second, it must bring into balance the treatment of continuous and digital systems, and provide all electrical engineering students with a background in digital systems comparable to that which they currently acquire in continuous systems. Third, it must result in a much wider and more effective use of the digital computer as a tool for system analysis and design in all engineering courses.

The COSINE Committee feels strongly that, as an essential first step, electrical engineering curricula should be made substantially more flexible. The movement toward greater flexibility is already under way in most engineering curricula; it is only in the climate of flexibility that engineering education can respond to the rapid advances in science and technology and adapt to the explosive growth in knowledge that is now taking place.

Because it is one of the fastest changing fields in natural sciences, and owing to its wide diversity of subject areas, the need for flexibility is particularly acute in electrical engineering. Apart from computer sciences, the fields of solid-state electronics, quantum and optical electronics, integrated circuits, bioelectronics, plasmas, control, communication, and large-scale power systems are but the more prominent of the many important subject areas that comprise electrical engineering. Each of these areas has its own needs and objectives, which, in many cases, cannot be satisfactorily met within the framework of a single curriculum with just a few electives and a large core of required courses in engineering and electrical engineering.

A climate of flexibility is thus essential for the accommodation of the needs of computer sciences as well as other subject areas within electrical engineering. It should be noted that some electrical engineering curricula already offer the student almost a full year of electives with more available by route of petition. In most cases, this approach is sufficient to enable a student to focus his studies in a field of concentration that may be either computer sciences, or computer sciences in combination with such areas as circuits, systems, control, or solid-state devices. The committee feels that flexibility of this order of magnitude would permit many diverse educational programs to thrive within electrical engineering and make it possible for students in such programs to acquire excellent training in both the foundation subjects in electrical engineering and the more specialized subjects in their particular fields of interest. Indeed, such training would serve well not only the needs of those students who would continue their studies toward higher degrees, but also of those who would terminate their formal education at the undergraduate level.

A computer science program in electrical engineering

By a computer science program in electrical engineering the committee means a curriculum in electrical engineering education that allows the student to acquire substantive competence in computer sciences and related fields, comparable but not necessarily similar in content to that acquired by students in a typical computer science department. Such a program must fulfill the following aims:

1. It must provide the student with a thorough understanding of computer systems and their use that is based on fundamental principles of long-term value rather than the salient facts of contemporary practice.

2. It must give the student a background in relevant discrete mathematics (set theory, mathematical logic, and algebra) including familiarity with methods of deduction as applied to abstract models relevant to the field of computation.

3. It must give the student access to a variety of subjects covering specialized and advanced aspects of computer science.

4. It must provide the student with sufficient technical and general knowledge that he can readily broaden his

This article is a condensation of a report, of the same title, issued by The Committee on Computer Sciences in Electrical Engineering (COSINE Committee) of the Commission on Engineering Education. Chaired by Dr. Samuel Seely, educational consultant, the committee includes J. B. Dennis (M.I.T.), D. C. Evans (University of Utah), W. H. Huggins (Johns Hopkins University), M. Karnaugh (IBM Corp.), J. F. Kaiser (Bell Telephone Labs.), F. F. Kuo (University of Hawaii), E. J. McCluskey (Stanford University), W. H. Surber and M. E. Van Valkenburg (Princeton University), and L. A. Zadeh (University of California). Copies of the report may be obtained by writing to the Commission on Engineering Education, Washington, D.C. 20036.

education through continuing study and remain adaptable to the changing demands of society throughout his professional life.

Our discussion relates to the first three of these objectives; achieving the last objective is left to the discretion of each university.

The committee recognizes the inherent difficulty in attempting to specify a detailed curriculum in computer science; no single curriculum could possibly fit into the variety of programs and organizational frameworks present in electrical engineering departments. We have, therefore, organized the material into *subject areas* as shown in Table I; each subject area is a collection of related topics having cohesion and purpose. In describing a subject area, the committee does not wish to imply that it necessarily corresponds to a single one-semester course. Furthermore, the description provided for each subject area is intended only to indicate what the committee regards as a reasonable set of topics and their logical order, without implying that strict adherence to the description is expected.

Category A comprises four subject areas that the committee feels are of central importance and basic to an adequate education in computer science. Other subject areas, which are less central but which nonetheless cover important related and specialized material, are listed in Category B. The committee feels that these subject areas should be available to students in the computer science program. However, we do not view Category B as necessarily complete, since there are legitimate differences of opinion on whether additional areas should be offered.

A computer science program in electrical engineering can assume a variety of forms. It can start at the freshman, the sophomore, or the junior level. It can be structured as an option with specified required courses, restricted electives, and unrestricted electives. It can be realized by allowing for enough electives in a standard electrical engineering curriculum to make it possible for a student (with the help of a faculty advisor) to put together a program of his own in computer sciences. It may or may not include a core of required electrical engineering courses in areas outside of the computer sciences. Accordingly, the subject outlines presented here should be regarded as guidelines intended to assist electrical engineering departments in devising curricula in computer sciences.

Finally, the committee takes no position on jurisdictional questions relating to departmental responsibility for particular courses. Because information processing in all of its forms will continue to be of major concern to electrical engineering departments, the committee feels that electrical engineering faculties should strive to develop strong expertise in computer sciences and related areas. At the same time, it is essential that electrical engineering departments cooperate closely with all departments having interests in computer sciences, sharing with them the responsibility for providing instruction in computer-oriented courses and for conducting research in computers and computer-related areas.

I. Subject areas for a computer science program in electrical engineering

Category A: Basic subject areas
- Programming principles
- Computation structures
- Introduction to discrete mathematics
- Machines, languages, and algorithms

Category B: Recommended elective subject areas
- Digital devices and circuits
- Switching theory and logical design
- Programming systems
- Operating systems
- Numerical methods
- Optimization techniques
- Circuit and system theory
- Information theory and coding
- Functional analysis
- Combinatorics and applications
- Probability and statistics
- Symbol manipulation and heuristic programming

Category A: Basic subject areas

We suppose that the student embarking on this program has had previous exposure to the use of automatic computing, whether in high school or in work experience. In some schools it might be desirable for students in computer science to take immediately the basic course: Programming Principles. In such cases, some arrangement to provide an early introduction to elementary numerical methods should be provided.

The four subject areas of Category A comprise material that is essential background for all students of computer science. Two subject areas, Programming Principles and Computation Structures, are intended to give students fundamental knowledge of the operation of general-purpose-computer systems and the important features of programming languages, with emphasis on computer hardware as the means of realizing programming features. The indicated sequence of development shows our preference for developing familiarity with programming features prior to the discussion of issues of machine organization, instruction, code design, and addressing mechanisms. In this manner, it is possible to motivate aspects of machine organization by the language features they serve to implement. This approach also places conventional machine organization in a less sacred light and should lead students to consider and evaluate alternative implementations.

The subject area labeled Introduction to Discrete Mathematics is intended to familiarize the student with mathematical concepts and techniques that are basic to the study of discrete systems. Such familiarity is essential for computer science majors, and certainly very desirable for all electrical engineering students.

The subject Machines, Languages, and Algorithms serves to introduce students to abstract formulations of certain important and related areas of knowledge concerning computation. These areas are not only important in their own right, but they give the student the background and experience that will enable him to make significant use of abstract modeling in his future professional work. Suggested content for these courses is considered in the following discussion.

Programming principles. A reasonable selection of topics might include practice in algorithm design and programming to provide familiarity with the primitive operations on commonly encountered data types—for example, truth values, integers, real numbers, arrays, symbol strings, queues, stacks, trees, and lists; infix

and polish notation for expressions and the use of a pushdown list for their intertranslation and evaluation; assignment operator, conditional expressions, iteration, and subscripting; programs as defining functions with certain domains and ranges; building complex programs (functions) through the composition (nesting) of more elementary routines: binding of arguments, local and global identifiers and their scopes, sharing, and recursion.

A formalism for defining the syntax of programming languages, such as the Backus-Naur Form (BNF), should be introduced and the notions of derivation and ambiguity treated.

Computation structures. A knowledge of logical design fundamentals is an essential component of this subject. Topics include the realization of Boolean functions by combinational gate logic, the flip-flop, registers as ordered sets of flip-flops, register transfer operations, and theory and design of sequential control logic. Basic topics on number representation and the implementation of arithmetic operations include the binary number system, representation of negative numbers, simple mechanizations of addition, multiplication, division, and floating-point representations of real numbers.

Introduction to discrete mathematics. A representative set of topics includes propositional logic, Boolean algebra, set-theoretic notation, axiom systems and formal deduction, formal and informal proofs, proof by contradiction and finite induction, quantification and its use in formalizing propositions, and application to the study of formal properties of number systems, graphs, fields, groups and semigroups, and linear transformations.

Machines, languages, and algorithms. The following is a representative selection of topics: the finite-state model, state diagram and flow-table descriptions, equivalent states, equivalent machines, state reduction, finite-state languages, regular expressions and Kleene's theorem, limitations of finite-state automata, formal languages, grammars and derivations, context-free languages and their relation to pushdown storage automata, ambiguity and other properties, and sentence-parsing procedures. Computability topics include Turing machines, universal Turing machines, the existence of noncomputable functions, the "busy beaver" and halting problems, unsolvable problems of practical interest (for example, undecidable properties of context-free languages), the computability of recursive functions, Post systems, and Church's thesis.

Category B: Elective subject areas

As already noted, the subjects in Category B are less central to a computer science program than those in Category A. Moreover, the following lists are not necessarily complete, and individual ideas can be reflected in this group.

Short descriptions of the subjects are given in order to indicate the content and level of the material. The subjects included may denote courses, and, in a few cases, they can represent more than one course.

Digital devices and circuits. Modeling of nonlinear circuit elements; approximate analysis of quiescent and transient circuit behavior; use of time-domain circuit simulation; designing to specification with imperfect components; worst case and statistical approaches to circuit reliability. Applications to flip-flops, multivibrators, and networks of cascaded gate circuits; signal transmission methods; integrated-circuit technology. Physical phenomena usable to realize memory functions; ferromagnetics, cryogenics, electrostatics, photochromic materials, sound waves; address selection principles (coordinate and serial). High-current switches for inductive loads; sense amplifier design; techniques for improving signal-to-noise performance.

Switching theory and logical design. Combinatorial logical design, including the notion of prime implicants. Huffman theory of sequential machines, both synchronous and asynchronous. Hazards and their resolution. Interconnection of submachines to form larger units. Time-independent logical design. Identification and diagnosing experiments. Error detecting and correcting codes. Languages for specifying digital systems.

Programming systems. Formal methods of specifying language syntax and semantics. Syntactic structure, parsing methodology, diagnostics. Advanced study of programming features—for example, data structures, properties of data types, block procedures and the context of identifiers, parallelism and sharing of data, protection and process monitoring features. Implementation questions, including symbol table structure, code optimization, efficient subscripting, flow-of-control analysis and loop organization, flow-of-control and loop optimization, subroutine linking and parameter passing, syntax-directed compiling.

Operating systems. Functions of an operating system, such as controlling the use of computer system resources by programs submitted for execution by its users and insuring the integrity and security of information held on behalf of users. Topics suitable for in-depth study include the concept of process, the blocking and awakening of processes, the meaning of interrupts, interprocess communication, and process scheduling; the concept of address space, binding of procedures and data to address space and interprogram linking, motivation for location-independent addressing and techniques of implementation, shared information; storage management aspects, such as movement of information within a storage hierarchy, file backup, and issues of data integrity on restart. File access control and transfer of access privilege.

Numerical methods. Solution of systems of linear equations (matrix inversion, gauss elimination, determinants, etc.), numerical solution of nonlinear algebraic equations, roots of polynomials, interpolation techniques and curve fitting, numerical integration, solution of ordinary differential equations, solution of partial differential equations, and linear programming.

Optimization techniques. Solution of linear inequalities, linear programming algorithms, convex sets and convex functions, nonlinear programming, quadratic programming, dynamic programming, gradient techniques, maximum principle, Markoffian decision process, optimization under vector-valued criteria, and search strategies.

Circuit and system theory. Circuits as interconnections of basic elements, including such topics as passive and active circuits, characterization of circuits in the time and frequency domains, solutions of differential input–output relations. State-space formulation and representation by differential and difference equations. Basic properties of linear systems, time-varying systems, and nonlinear systems. Controllability, observability, and stability.

Information theory and coding. Quantitative definition and measurement of information, entropy of uncertainty, memoryless discrete channel, capacity of a memoryless channel, capacity theorems. Encoding and decoding of messages, parity check codes, convolutional encoders and decoders, sequential coding. Practical digital communication systems.

Functional analysis. Functions, functionals, and operators. Metric and topological spaces, linear spaces, Hilbert spaces. Linear functionals, differentiation of abstract functions, homogeneous forms, and polynomials. Stationary problems, fixed point theorems, gradient techniques. Quasi-linearization. Applications to problems in optimization and identification.

Combinatorics and applications. Enumeration techniques, including permutations and combinations, generating functions, recurrence relations, the principle of inclusion and exclusion, Polya's theory of counting. Theory of graphs, including planar graphs and duality. Network flow problems and elementary linear programming.

Probability and statistics. The concept of sample space and random variables, probability distributions on discrete sample spaces, dependent and independent random variables, conditional distributions, distributions on continuous sample spaces, parameters of probability distributions, normal distributions, stochastic processes. Markoff chains, waiting-line and servicing problems, estimation techniques, stochastic approximations, and decision rules.

Symbol manipulation and heuristic programming. Heuristic versus algorithmic methods, LISP and other relevant programming methods, game-playing programs, question–answer programs, symbolic integration and differentiation, theorem proving, search techniques, simulation of learning and concept formation, applications to pattern recognition and information retrieval.

Table II places the foregoing discussion into the context of a four-year undergraduate program. This table contains only the basic courses in Category A plus an introductory course in Programming and Numerical Methods. Detailed curricula of two electrical engineering departments, which show how they have included a concentration in computer sciences, are given in Appendixes A and B.

Implications of the digitalization of information processing

During the past two decades, as a result of the invention and development of a number of electronic components, circuits, and devices—such as the transistor, the magnetic-core memory, integrated circuits, etc.—it has become practicable and economical to process large volumes of data in digital form with high speed, accuracy, and reliability. We have witnessed a rapidly growing trend toward the use of digital systems in place of analog or continuous systems for purposes of computation, information processing, and control. Moreover, as a result of the availability of efficient, economical and reliable digital devices, modern information processing and control technology is becoming increasingly digital in nature, with all signs pointing toward a much bigger role for digital as compared with analog systems in the years ahead.

The transition from the analog and the continuous to the digital and the discrete has not yet been adequately reflected in the orientation of electrical engineering curricula. Many, or perhaps most, electrical engineering departments still lay a heavy stress on courses in continuous (in time, amplitude, and state) systems and devices, disregarding the fact that such courses are much less relevant to the needs of present technology, and certainly much less relevant to the needs of the future than they were 20 years ago, in the age of the vacuum tube and the amplidyne. The committee strongly feels that, in this regard, electrical engineering curricula are in need of a basic reorientation from the entirely analog and the continuous to reflect the digital and the discrete, and that electrical engineering departments should make a concentrated effort to prepare their students to deal with digital systems, be they computers, control systems, or special-purpose information- and data-processing systems.

How can such a reorientation be implemented? Clearly,

II. Skeleton of program showing recommended courses in computer science

Year	Term 1	Term 2
Freshman	...	Programming and numerical methods
Sophomore	...	...
Junior	Programming principles	Computation structures
Senior	Machines, languages, and algorithms	...

a wide-ranging shift in emphasis from the continuous to the discrete in electrical engineering curricula would present formidable problems that are not likely to be solved quickly or painlessly. Deeply entrenched attitudes will have to be changed, new knowledge and skills will have to be acquired, and new textbooks will have to be written. Indeed, it is beyond the scope of this report to analyze these problems fully and to suggest possible solutions to them. Thus, in what follows, the committee will restrict itself to making a few preliminary recommendations that suggest evolutionary changes in electrical engineering curricula. This section will discuss the addition of three new courses dealing wholly or in part with some of the basic aspects of discrete systems. The next section will discuss the digital reorientation of a number of existing courses.

New-course development. To provide a start toward the development of new courses that have a discrete state orientation, we discuss three possible courses as indicative of the direction that such development might take.

Course 1. Our first recommendation is that serious consideration be given to the development of a sophomore- or junior-level introductory course in circuits, systems, and signals, which would cover the fundamentals of both discrete and continuous systems. We envisage that, initially, such a course would be offered as an alternative to the traditional type of course in which the emphasis is wholly on the techniques of time- and frequency-domain analyses of linear, time-invariant, continuous-time circuits and systems. Eventually, courses of this new type would probably replace introductory courses of the more conventional nature.

An example of the type of course being recommended is that being developed at M.I.T. by Professors Athans, Dertouzos, and Mason, under the title, "Elements, Systems and Computation." In addition to covering the basic techniques of the analysis of linear, time-invariant, lumped-parameter networks and systems, this course also treats basic techniques for the study of nonlinear and discrete-state systems, and covers computational as well as analytic methods of problem solving in the context of such systems.

A controversial aspect of a course of this type is that its broader coverage of both continuous and discrete systems is attained necessarily at the cost of less depth in the treatment of various types of components and systems. For this reason, it may be preferable, in the longer run, to treat discrete systems separately in a course that would precede a course in continuous systems. Although this approach would represent a departure from the traditional order, it may well be more logical and more sound pedagogically.

Course 2. Our second recommendation relates to the inclusion of a new type of course, at the junior or senior level, that would be concerned chiefly with mathematical concepts and techniques central to the analysis and synthesis of discrete, as contrasted with continuous, systems. This course is listed in Table I as Introduction to Discrete Mathematics.

A representative set of subjects that might be included in a course of this type would include elements of set theory; Boolean algebra; elements of mathematical logic; elements of the theory of relations, groups, fields, and rings; elements of Galois theory; etc. A course of this type would serve essentially the same function in relation to discrete systems that the conventional courses in Laplace transforms, complex variables, linear algebra, etc., serve in relation to the analysis of linear time-invariant systems. Clearly, of course, the totality of the mathematical background needed for the analysis and synthesis of discrete systems cannot be provided in a single course.

A theoretically oriented student majoring in computer sciences might well take one or more courses in mathematics, in such subjects as abstract algebra, set theory, mathematical logic, group theory, etc., in preference to taking a single, less-specialized course of the type here being discussed. Thus, the committee's recommendation is intended primarily to point to a need in electrical engineering curricula for a broadly based course in the mathematics of discrete systems, which would be suitable for most electrical engineering students, not just for those majoring in computer science. A desirable first step in this direction may be accomplished by a revision of the usual two-year mathematics program that exists in almost all electrical engineering curricula (the calculus program), to a program of which one half is devoted largely to discrete mathematics and the second half to topics in continuous mathematics.

Course 3. Our third recommendation relates to the offering of a course in finite-state systems at the junior or senior level. The importance of such a course stems from the fact that finite-state systems constitute a very basic class of systems particularly well-suited for the introduction of such basic concepts as state, equivalence, identification, decomposition, etc. Furthermore, they are much better suited for computational purposes than continuous systems, and can frequently be used as approximate models for the latter. At present, several electrical engineering departments offer courses of this type, covering such topics as the characterization of finite-state systems, the notions of state and system equivalence, identification algorithms, decomposition techniques, synthesis techniques, etc.

Until a few years ago, the offering of courses on finite-state systems was hampered by the dearth of texts on this subject, a situation that is now changing. There are several very good undergraduate level texts on finite-state systems. The teaching of a course on this subject should be a relatively easy task for most electrical engineering professors. Consequently, the committee feels that every electrical engineering department should consider offering an elective course on finite-state systems as part of its regular curriculum.

Implementation

To introduce computer techniques in a meaningful way into traditional courses covering circuit theory, control, communication systems, and similar topics, the following sequence might be used in presenting subject matter: theory, analytical methods of solution, numerical algorithms, and computational examples. Some general recommendations are outlined in the following; these recommendations presuppose a familiarity with programming principles and elementary numerical methods.

1. The first course in electrical engineering (usually in the sophomore year) should be modified to incorporate the use of computers as a tool. A problem-oriented program having a special language might be used to work exercises relatively early in the course, without requiring

a detailed knowledge of computer programming.

2. Major revisions might be made in the method of presentation for certain courses, particularly those in the systems area. For programs in control theory and communication systems, for example, this approach might involve changes in emphasis of some of the traditional material and the introduction of new material related to computer operation and limitations.

3. Since the purpose of courses in the systems area is to develop an understanding of the behavior of these systems, it would be desirable to make available to the students as analysis and design aids certain fairly elaborate specialized computer programs with provision for graphic output. Computer-generated results would thus be provided without the large investment of the student's time that would be required were he forced to write and debug all of the necessary programs.

4. The academic program should help develop a more thorough understanding of programming techniques and the limitations of numerical methods of simulation, particularly in those areas in which the computer is a major factor in the practice of modern engineering, either as a design tool or as an important part of the system.

5. The use of computers for system simulation should be encouraged as a valuable supplement to laboratory experiments with physical elements.

Computer use for simulated laboratory experiments

Digital computer simulation of devices and systems can provide a valuable supplement to laboratory experiments with physical elements. Simulation studies, provided that adequate software is available, can also be used very effectively as alternatives to some problem sessions and homework exercises. It is recommended that such computer experiments on idealized models of physical devices be introduced into the laboratory program early in the curriculum. The essential features of the behavior of many types of systems can sometimes be explored more readily in this way than by actual experiments.

Idealized models can, of course, be nonlinear and thus can provide quite realistic representations of the true device characteristics. Some advantages of a computer study are the ease with which the model parameters can be varied over a wide range without damage to the components, the ability to compute sensitivity coefficients and make a worst-case analysis, and the ability to generate and plot performance curves directly for nonlinear as well as linear systems without tedious experimental tests or the drudgery of repeated hand computations. Furthermore, this approach allows a considerable degree of individual initiative to be exercised by the student in the design of the system model to be simulated.

Observation of the behavior of the actual physical system is very important, and experiments on real devices and systems should clearly be retained. Both types of experiments are significant in different ways in helping to develop an intuitive feeling for system behavior; a balance should be maintained between actual and simulated experiments. A combination of both will be more stimulating than either type alone, and has the added advantage of providing a basis for appreciating the differences between the analysis of an idealized model and the behavior of the real device.

For example, digital simulation of models of electronic circuits can be a significant adjunct to normal electronics laboratory experiments. A number of quite elaborate electronic circuits analysis computer programs are available or are being developed. Although many of these have serious deficiencies from an educational viewpoint, there is some expectation that this will be rectified in the relatively near future. An electronic circuit simulator, such as ECAP, when supplemented by a graphical output routine can provide the basis for some very useful simulation studies in this field. Examples might include the study of transistor amplifier operating-point stability with respect to parameter variations, the transient and frequency response characteristics of pulse amplifiers, and oscillation phenomena.

Educational software

The most desirable types of specialized software that should be available in the computer library can be divided into several general categories:

1. Mathematical subroutines. Examples include graphic output subroutines and programs to solve simultaneous sets of linear equations, to find the roots of polynomials, and to solve sets of ordinary differential equations. These subroutines are presently available at almost all computer centers. They are far from being sufficient, however.

2. More elaborate programs to facilitate special types of analysis. Examples are frequency response, root-locus and transient-solution plotting, Fourier-series analysis routines, parameter optimization routines, and some statistical analysis routines.

3. Problem-oriented programs with special language facilities. Examples of such programs are analog system simulators, such as MIMIC and CSMP; digital system simulators, such as BLODI; and electronic circuit simulators, such as ECAP. Another significant example is JOBSHOP, which was developed by W. H. Huggins at The Johns Hopkins University, as a simulator for the circuit design process. Much remains to be done in the development of suitable software for educational purposes. In addition, information concerning the availability of the computer programs that do exist, and their documentation, leaves a great deal to be desired.

Closing comments

The recommendations presented in this article relate to what the committee believes are the central issues in the impact of computers and computer sciences on electrical engineering education. These are (1) the need for computer science programs in electrical engineering, (2) the need for greater emphasis on discrete systems in electrical engineering curricula, and (3) the need for modifying the content and underlying philosophy of basic electrical engineering courses, particularly in circuits and systems, to interweave the use of computers for analysis and design with the development of basic theory.

These issues are probably the most pressing of the many questions and problems facing electrical engineering education today. They are by no means the only issues arising out of the advent of the computer age; clearly, the use of computers will have to be stressed not only in courses in circuits and systems but, more generally, in all areas in electrical engineering in which computers can be an effective tool for analysis, design, or simulation. We have said nothing concerning the roughly three years of studies constituting the portion of the B.S. program that reflects the general base of electrical engineering. However, we do stress that attention must be given to the revision of introductory courses in mathematics, physics, and other basic fields, with a view to increasing the emphasis on algorithmic and numerical techniques in such courses. Also, the traditional role of laboratory courses must be re-examined, in the light of the possibility of using computers as simulators of physical systems. Already, in many instances, greater insight into system behavior may be obtained by studying its performance with the aid of a computer than by measuring the physical variables and parameters associated with it. This will be even more true in the future.

It hardly needs saying that computers and computer sciences are, and will be, of considerable concern to many disciplines in addition to electrical engineering, and electrical engineering departments will have to cooperate closely with other academic departments, especially computer sciences and mathematics departments, both in instruction and in research in computers and related areas.

During our study, we have maintained liaison with the ACM Curriculum Committee on Computer Sciences through E. J. McCluskey, and we are especially indebted to William Viavant of that committee for his assistance. We also wish to acknowledge the support of our efforts by the National Science Foundation through the Commission on Engineering Education, and its executive director, Newman A. Hall.

Appendix: Some electrical engineering curricula that include computer science courses

A. College of Engineering, University of California, Berkeley

B.S. Computer Science Program

Year and Course	Quarter Hours
Freshman	
Mathematics	12
Chemistry	12
Physics	7
Electives[a]	14
Sophomore	
Mathematics	12
Physics	12
Electives[a]	21
Junior	
Electric circuits	8
Electronic circuits	5
Electronics and circuits laboratory	6
Linear systems analysis	4
Computers and information processing	4
Restricted[b] and technical[c] electives / Humanistic–social[d]	18
Senior	
Switching and computer circuits	6
Digital computer systems	7
Laboratory	2
Restricted[b] and technical[c] electives[b] / Humanistic–social[d]	30
Total:	180

[a] The electives for the freshman and sophomore years include at least 15 hours of humanities or social sciences, plus
Computers and their applications 4
Introduction to electronic systems, circuits, and devices 4
Engineering mechanics 4
Properties of materials

[b] Restricted electives: three courses from an available list, including mathematics, physics, engineering courses.

[c] Technical elective: 25 units of upper division computer science, engineering, mathematics, physics, statistics, or other natural science courses.

[d] Humanistic–social: Total in the program must meet minimum College requirements.

B. Department of Electrical Engineering, Massachusetts Institute of Technology

S.B. in Electrical Engineering,[a] Computer Science Program

Year and Course	Credit Hours[e]
Freshman	
Calculus	24
Chemistry	12
Physics	24
Introduction to automatic computation	6
Humanities	18
Elective	6
Sophomore	
Physics[b]	12
Physics	12
Elements, systems, and computation[b]	12
Elements, systems, and computation	12
Programming linguistics[c]	12
Elective[b]	12
Humanities	18
Junior	
Circuits, signals and systems	12
Electromagnetic fields and energy	12
Computation structures[c]	12
Computer systems[c]	12
Electives[d]	24
Humanities	18
Senior	
Electives[d]	60
Humanities	18
Thesis	12
Total:	360

[a] An unofficial curriculum presently under consideration.

[b] Subjects to meet an M.I.T. "science distribution requirement."

[c] Basic computer science courses now under development.

[d] A variety of suitable computer science elective subjects is currently available.

[e] Three credit hours represent approximately one semester contact hour.