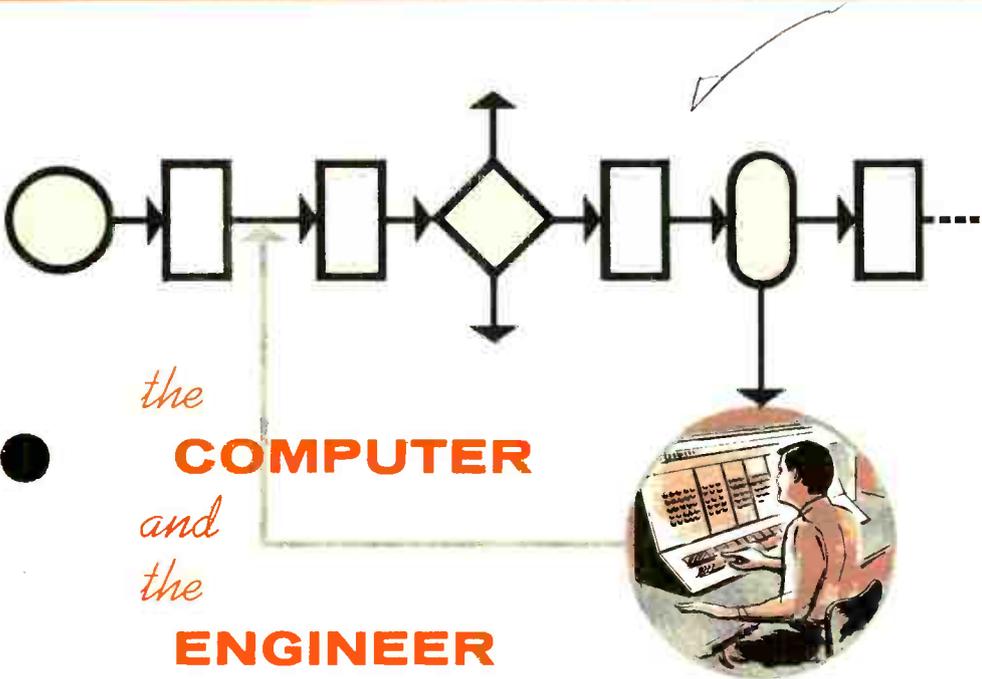


the *Lenkurt*[®]

Demodulator



The computer has become a most valuable and necessary piece of equipment in the design laboratory of today's electronics firm. Through accuracy and speed in complicated mathematical calculations, the computer has made possible design techniques never before feasible.

This article concerns the scientific digital computer and its uses by the electronics engineer.

In a relatively short life, computers have grown out of desk-top calculating machines, relying on a complicated assortment of gears and mechanical linkage to perform their task, to electronic giants capable of thousands of mathematical calculations a second. Computers are called on for mammoth data processing duties, for controlling automated machines and, of prime interest to the electronics design engineer, to carry out complicated mathematical computations with a speed and accuracy never before possible.

Computer Types

Computers are of two basic types, analog and digital. There is very little similarity between the two, and they serve markedly different functions. The analog computer is used primarily as an analysis device, where the physical properties of a system—mechanical, electrical, or whatever—can be represented by analogous circuits in the computer. With amplifiers, potentiometers, and circuits performing such mathematical functions as differentiation and integration, the computer imitates the real item. Once an electrical model is devised, analyzing the system is particularly suited to the analog computer.

In operation, the system is tested under varying conditions by changing voltages in the computer, with all calcula-

tions carried out simultaneously. Results are displayed immediately on one of a number of readout devices: strip-chart recorders, X-Y plotters, or even oscilloscopes. The complexity of the model has little to do with the computer's problem-solving speed.

Digital computation, a more time consuming step-by-step process, is much more accurate, and is limited basically only by the preciseness of input data and the ability of the machine to store and use significant figures. Digital computers, with which this article is primarily concerned, are basically manipulators of stored information.

The computer (Figure 1) consists of a central processing unit, where the actual computation is done, a storage unit providing the machine with an electronic memory, and input and output devices with which the user can communicate with the machine.

The facts of a problem are held by the machine's memory — commonly a large grouping of magnetic cores which can retain thousands of single electrical pulses for later use. Instructions to the computer, called a *program*, are written by the operator to guide the machine in processing and changing the stored pieces of information in a prescribed way. In data processing operations, huge quantities of information are handled, facts sorted, relatively easy computa-



Figure 1. Typical scientific computer is the IBM 1620 II, used at Lenkurt. Behind the central processing unit is magnetic disc storage drive.

tions done, and the information re-assembled in meaningful form — a printed page, a stack of punched cards, payroll checks, or perhaps utility statements.

Scientific Use

Scientific computers serve an entirely different function. The computer first sees the scientific or engineering problem as a small set of numerical facts; the answer will probably be no more

than a short series of digits. But inside the computer an enormous amount of work has gone on. Mathematical computations which might have taken weeks — if possible at all — had they been done manually by the engineer, are now obtained from the computer in minutes.

The digital computer does all its work by breaking down complex mathematical functions into many simplified arithmetic operations, much as the me-

chanical desk calculator does multiplication by a series of additions. With the computer, difficult mathematical problems can be reduced to processes of addition, subtraction, and logical decisions, performed thousands of times per second.

The theoretical knowledge necessary to construct a computer had been available for many years, but it was not until reliable electronic components freed the computer from the limitations of the mechanical calculating machine that the "brain" began to grow.

The first computers were merely arithmetical manipulators able to do a sequence of calculations without further human control. They were complex, could handle huge quantities of data, and did it with speed never before imagined possible. But they were tied to limited specific processes, much like a platoon of girls operating desk calculators with instructions to carry out a single set of arithmetic operations and list them on a form.

Logic Added

An historic hurdle was overcome when logic functions were added — the computer now had the ability to make decisions. This is synonymous to having the desk calculator operators stop at some point in their work, analyze the figures, and proceed on one of a number of alternate paths. In its simplest form, logical response is to answer "yes" or "no" to a certain stimulus; that is, to take one of two possible paths. Judging the presence or absence of some variable condition, and making a decision based on this, is the function of the elementary logic circuit of the computer. In this

way the modern computer makes decisions which will determine some later course of action.

Beyond the ability to make decisions comes the faculty of *learning* from experience. Some computers can analyze their own method of approaching a problem and improve on it the next time the same problem is presented.

The availability of electronic computers in the early 1950's for use by design engineers greatly relieved the tedious routines followed until that time. One of the common — and laborious — problems faced by telecommunications equipment designers is the creation of new filter networks. (About 90 percent of *all* networks are filters, defined as circuits to separate wanted signals from groups of signals.) Well-developed methods are available for making accurate mathematical models of network circuits, involving extensive use of complicated calculations with

32	16	8	4	2	1	NUMBER
○	○	○	○	○		1
○	○	○	○		○	2
○	○	○	○			3
○	○	○		○		5
○	○		○		○	10
	○		○			43

Figure 2. Sample of binary number equivalents used in computers. Values increase in multiples of two, right to left.



Figure 3. Electric typewriter allows engineer to supply data to computer as design problem progresses. Results of computation also appear on the typewriter.

many repetitive functions. This suits the digital computer perfectly.

Binary System

The digital computer, as opposed to the analog machine, deals with discrete numbers. Just as the mechanical calculator relies on the exact positioning of gears for computation, the computer positions binary pulses in a precise method, storing them in its electronic memory until they are to be used.

The binary number system in computer machine language differs from conventional decimal numbers only in concept. In customary numerical expression using the decimal (or more correctly the "denary") system, a number such as 837 is merely a convenient method of listing the number of units, tens, hundreds, etc. In each position there is a possibility of 10 different integers, 0 through 9. The number 43 really means $(4 \times 10^1) + (3 \times 10^0)$.

The binary system is based on the

possibility of only two choices in each position, written as either 1 or 0. Examples are given in Figure 2. The number 43 is written 101011, or $(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$. While this may seem a cumbersome notation, it is very appropriate for digital computers because it corresponds with the natural two-state ability of many electronic components—relays, magnetic core memory units, and flip-flop circuits.

Computer Languages

To communicate with the computer, the user must either *talk* in the basic machine language or, if he prefers, provide a means of translation in some other language. In modern computers, several programming languages are available to the user, depending on the specific purpose at hand. Special "compiler" programs already in the computer perform the translation. In this way the programming languages are *user ori-*

ented, rather than machine oriented, and allow the programmer freedom of expression and thought in terms more familiar to him.

A language developed primarily for scientific work is FORTRAN (FORMula TRANslation) which bears a close resemblance to the native tongue of the engineer and scientist — mathematics. A similar language more popular in Europe is ALGOL (ALGebraically Oriented Language). Add to these COBOL (COMmon Business Oriented Language), PL/1 (Programming Language/1), and a growing number of languages for specific disciplines.

The facts of the scientific problem, along with the program to be used to reach a solution, are fed to the computer from one of a number of input devices — magnetic tape, magnetic discs, punched cards, perforated tape, and the typewriter. The computer stores this information in its electronic memory for future use. Certain facts may be supplied directly by the engineer, operating an electric typewriter at the computer console (Figure 3). Answers also will appear on the typewriter when the problem is completed.

When a program is first written for a problem, a flow diagram is constructed expressing the logic of the approach. A non-mathematical example in Figure 4 illustrates decision making, computer style.

Iteration

The example also points to the repetitive ability of the digital computer, one of the best used qualities of the computer in scientific calculation. By the mathematical process of *iteration* (or

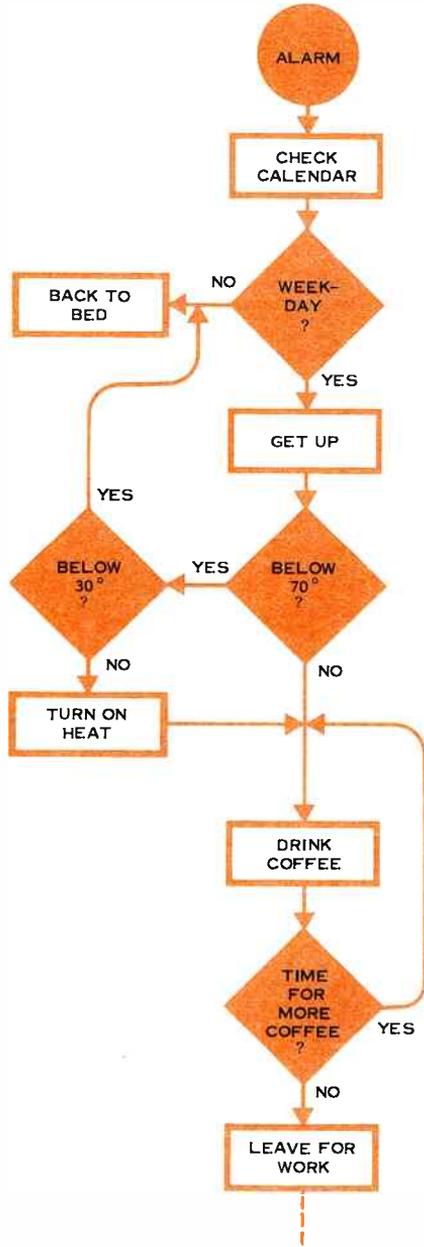


Figure 4. Simulated flow diagram illustrates logic used in planning computer program.

repetition) the computer can begin with a guess at the answer and then check it. If necessary, the computer will modify and repeat the original calculation. With each iteration, the machine answer approaches the correct value sought.

One very common example of an iteration, known as the Newton method, is illustrated graphically in Figure 5. The problem is to find the special value of a , the point where the curve crosses the x axis. The point also corresponds to $y = 0$. The computer makes a first guess x_1 for the value of a and then computes y_1 . Since y_1 is not zero, the computer seeks a better approximation of a . It computes the slope (or tangent) of the curve at (x_1, y_1) , then calculates where this straight line intersects the x axis. This value, x_2 , is taken as a better approximation of a . The sequence is repeated until the value of a is known as accurately as the capabilities of the machine will allow. The iteration is then stopped.

Another example of iteration is the solution of the quadratic equation

$$x^2 + ax - b = 0$$

This can be rearranged to give

$$x(x + a) - b = 0$$

$$x = \frac{b}{x + a}, \text{ and finally}$$

$$x_{n+1} = \frac{b}{x_n + a}.$$

With $n = 0$, a guess for the value of x is entered and the calculation is carried out for the right side of the equation. If the answer is equal to x , our guess has been accurate. If not, the first answer is substituted for x_1 , and the calculation

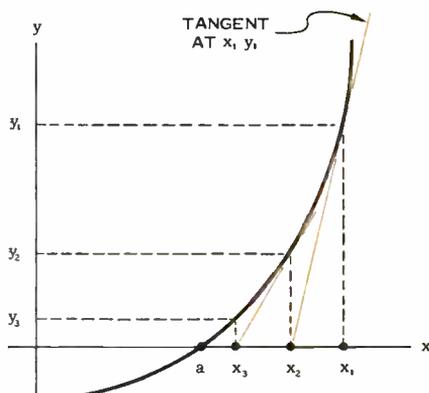


Figure 5. Computer uses iteration to find the point where curve crosses x axis.

is carried out again — approaching the true value of x at each iteration.

Given the equation

$$x^2 + 6.3x - 14.8 = 0,$$

the iteration approach would proceed in this way:

$$\text{Let } x_0 = 0$$

$$x_1 = \frac{14.8}{0 + 6.3} = 2.349$$

$$x_2 = \frac{14.8}{2.349 + 6.3} = 1.711$$

$$x_3 = \frac{14.8}{1.711 + 6.3} = 1.847$$

$$x_4 = \frac{14.8}{1.847 + 6.3} = 1.817$$

$$x_5 = \frac{14.8}{1.817 + 6.3} = 1.823$$

$$x_6 = \frac{14.8}{1.823 + 6.3} = 1.822$$

$$x_7 = \frac{14.8}{1.822 + 6.3} = 1.822$$

The iteration, in this case, was stopped when four significant figures in the answer stabilized. This process, by its nature, will converge on the smaller of the two roots of the equation. The second root is found immediately by dividing the final term in the equation by the first root:

$$\frac{-14.8}{1.822} = -8.122$$

Monte Carlo Method

Another important mathematical practice appropriate to the iterative ability of the computer deals with problems of probability, and is called the Monte Carlo method. In the manufacture of electronics equipment, component values vary randomly within the specified tolerance range assigned. With the help of the computer it is possible to calculate what percentage of finished products will be within a nominal performance range. Using the Monte Carlo method, the computer generates values for the circuit components in a random manner, much the way a factory worker would select actual parts during assembly. Each time different component values are selected, the circuit performance is checked by the computer. From this, a distribution curve can be plotted, indicating the probable number of acceptable units. If the percentage is too low, tolerances must be tightened for components most likely to affect the circuit, and the process is repeated. In this way the design engineer can use his mathematical model of the circuit to advantage, predicting with great accuracy the performance of the finished units leaving the assembly line.

Filter Design

A typical filter network design problem will illustrate the procedure used by the engineer. A standard network and its loss-frequency curve are shown in Figure 6. The designer will work with a number of computer programs, each intended to help him solve particular problems and approach a practical end product. The first program used will aid him in selecting the optimum parameters defining the performance of the filter. After feeding the program into the computer's memory, the engineer enters tentative values for certain design characteristics, such as passband edge frequencies (f_1, f_2), the frequencies at the infinite loss points ($f_{1\infty}, f_{2\infty}, f_{3\infty}, f_{4\infty}$) and the passband ripple. Then, one by one, the designer supplies the computer with sample frequencies in the stopbands and passbands. The computer calculates the loss at each of these points from which an accurate loss curve can be plotted.

When satisfied with his basic design, the engineer must determine the components to be used. A second program is placed in the computer's memory. Using what is called *insertion-loss theory*, the engineer and the computer can now generate the component values needed to satisfy the original design characteristics.

The results of this second operation are used in a general analysis program to prove the design under practical conditions (till now the components were considered to have pure capacitance and inductance). The computer then produces values for total loss, phase shift, envelope delay, reflection coefficient and input impedance.

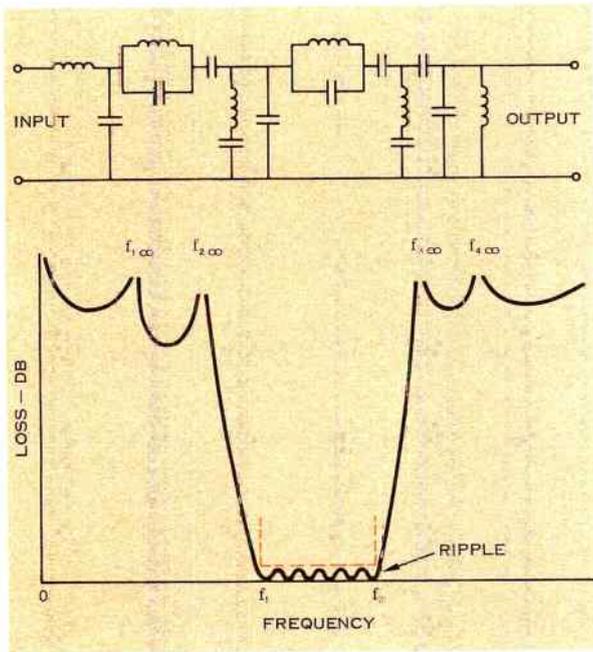


Figure 6. Filter circuit (top) will produce this type of loss-frequency curve. Engineer and computer determine optimum component values to provide proper passband frequencies, acceptable ripple, and infinite loss points.

A fourth computer step is used to examine tolerances, using the Monte Carlo method previously described. When all the design programs have been completed, a prototype model can be built, and the filter circuit is well on its way to becoming part of a new product.

Prior to the time when engineers had the work-horse capabilities of the computer at their disposal, a method known as *image-parameter* theory was used in filter design. Insertion-loss theory was just not practicable due to the vast amount of calculation necessary. But with the computer a slightly superior product can be produced using the insertion-loss technique, with the added advantage of the method's simplicity and straightforward approach.

Time Saving

Approximate times for performing the various steps may be listed, considering computers now found in most engineering facilities, like the IBM 1620II used at Lenkurt, but understanding that more modern and likewise faster machines are being produced constantly. The first program, used to select the basic performance parameters for a fairly complicated filter, takes the engineer and the computer together about 30 minutes. Component values are produced from the second program in about 10 minutes. The general analysis program takes an additional 10 to 15 minutes. Longest because of its many repetitive steps is the Monte Carlo tolerance analysis, running about 60 minutes. Total time: less than two hours.

The image-parameter method is not only slower by two to ten times depending on the complexity of the filter, but lacks the complete information obtained by the insertion-loss technique. In image-parameter design it is still necessary to construct a "breadboard" model to check dissipation characteristics and tolerances — all done on the computer with insertion-loss methods.

Other Uses

In addition to the design of networks, a great deal of time on the computer is spent analyzing one-time mathematical problems that do not lend themselves to generalized programming. Again, time-saving benefits of the computer are invaluable to the design engineer. As long as mathematics is a part of electronics design, computers will be a tool for the design engineer. And even today's methods, superior by many orders of magnitude to the labors of only a few years ago, will be supplanted as more sophisticated techniques are created, and man and machine learn better how to work together.

Probably one of the most promising developments in the last year is graphical on-line simulation of electronic circuits. Using a "light pen" the designer can draw his circuit on the face of a computer-coupled cathode ray tube. With proper commands to the machine's memory, he can analyze circuit

functions under specific parameters. Using the light pen, almost instantaneous modifications are possible, allowing the engineer to redesign the circuit and see the results while still at the computer.

The same device is used in other disciplines to analyze the structure of bridges, buildings, and even three-dimensional mathematical representations — viewed from side, top, or bottom.

Time Sharing

There is a trend to centralize computer facilities for use by a number of subscribers on a time-sharing basis. Several large computer centers have already been established around the country, with users connected remotely through telephone communications facilities. Time-sharing computers can allow two-dozen or more users on the line at one time, efficiently sandwiching their programs together with such speed that the user seldom, if ever, notices a delay. The user gains the advantages of large computer capability, while only paying for the computing time he needs.

The scientific computer, modern-day design tool of the electronics engineer, will improve in its usefulness as quickly as man's inventiveness will allow. Along the way, the computer is not only absorbing a tremendous work burden, but is making possible increased quality and better performance in electronics equipment.

HERTZ?

In an effort to reach worldwide understanding in published scientific and technical work, standardization of terms is essential. The Institute of Electrical and Electronics Engineers (IEEE) standards committee recently accepted a number of new standards of electrical units and symbols, established in close cooperation with many international organizations. At the beginning of the year the *Demodulator* adopted the IEEE recommendations.

One of the most noticeable changes is the adoption of the name *hertz* as a unit of frequency. While *cycles per second* is still widely used, and technically correct, *hertz* is now preferred in that it is more understandable in all languages.

In general, symbols for units are written in lowercase letters, except when the name of the unit is derived from a proper name. Thus, it is *hertz*, but *Hz* when abbreviated; *decibel*, but *dB*.

Compound prefixes are to be avoided: $\mu\mu\text{F}$ (micromicrofarad) becomes pF (picofarad); $m\mu\text{s}$ is now ns (nanosecond); and the familiar kmc (kilomegacycle) should be GHz (gigahertz).

Prefixes for Metric Units

10^{12}	tera	T
10^9	giga	G
10^6	mega	M
10^3	kilo	k
10^2	hecto	h
10	deka	da
10^{-1}	deci	d
10^{-2}	centi	c
10^{-3}	milli	m
10^{-6}	micro	μ
10^{-9}	nano	n
10^{-12}	pico	p
10^{-15}	femto	f
10^{-18}	atto	a

Examples of Usage

decibel	dB
gigahertz	GHz
hertz	Hz
(cycles/second)	
kilohertz	kHz
megahertz	MHz
picofarad	pF
siemens	S ($1\text{ S} = 1/\Omega$)
(mho)	

Readers may consult the IEEE Standard Symbols for Units (No. 260, January 1965) and other IEEE documents for more complete details.

Lenkurt Electric Co., Inc.
San Carlos, California

Bulk Rate
U.S. Postage

Paid

San Carlos, Calif.
Permit No. 37

MR. E. A. GARCIA
CHIEF OPERATOR
% MILWAUKEE ROAD DEPOT
SAVANNA, ILL.

R-24948 6B

RETURN REQUESTED

ALL-WEATHER CABINETS

Weatherproof . . .

Climate
Controlled . . .

Durable . . .

Standard
10' Racks . . .

Economical . . .



Lenkurt's 565 weatherproof cabinets offer complete protection of telecommunication equipment in all outdoor installations. Temperature and moisture control, insulation, and rugged construction make the 565 cabinet a comfortable home for your equipment in any climate, on pole or pedestal.

Factory installation of your new terminal and repeater equipment in the 565 is economical, too. Write Lenkurt, Dept. B720, for details.

LENKURT ELECTRIC

SUBSIDIARY OF
GENERAL TELEPHONE & ELECTRONICS GTE

San Carlos, California, U.S.A.

Lenkurt Offices

San Carlos	New York City
Atlanta	Washington, D. C.
Dallas	Chicago

THE LENKURT DEMODULATOR is a monthly periodical circulated to technicians, engineers, and managers employed by companies or government agencies who use and operate communications systems, and to educational institutions. Job title, company affiliation, and nature of business must be indicated on subscription requests. Each issue features an instructive article on a subject dealing with the science of telecommunications. Permission to reprint articles may be obtained by writing THE EDITOR.

World Radio History