Computing Science Technical Report No. 99
A History of Computing Research*
at Bell Laboratories (1937-1975)

Bernard D. Holbrook
W. Stanley Brown

1. INTRODUCTION

Basically there are two varieties of modern electrical computers, analog and digital, corresponding respectively to the much older slide rule and abacus. Analog computers deal with continuous information, such as real numbers and waveforms, while digital computers handle discrete information, such as letters and digits. An analog computer is limited to the approximate solution of mathematical problems for which a physical analog can be found, while a digital computer can carry out any precisely specified logical procedure on any symbolic information, and can, in principle, obtain numerical results to any desired accuracy. For these reasons, digital computers have become the focal point of modern computer science, although analog computing facilities remain of great importance, particularly for specialized applications.

It is no accident that Bell Labs was deeply involved with the origins of both analog and digital computers, since it was fundamentally concerned with the principles and processes of electrical communication. Electrical analog computation is based on the classic technology of telephone transmission, and digital computation on that of telephone switching. Moreover, Bell Labs found itself, by the early 1930s, with a rapidly growing load of design calculations. These calculations were performed in part with slide rules and, mainly, with desk calculators. The magnitude of this load of very tedious routine computation and the necessity of carefully checking it indicated a need for new methods. The result of this need was a request in 1928 from a design department, heavily burdened with calculations on complex numbers, to the Mathematical Research Department for suggestions as to possible improvements in computational methods. At that time, however, no useful suggestions could be made.

2. EXPEDIENTS FOR COMPUTING WITHOUT COMPUTERS

By 1928, the Bell Labs accounting department was making extensive use of punched-card equipment for cost accounting. This punched-card equipment was, from time to time, used by technical departments with extensive statistical jobs; in addition, members of the Mathematical Research Department made valiant efforts to use it for more purely mathematical problems, but with very little success. The then-available logical capabilities of punched-card equipment were too limited even for such tasks.

In some cases, the necessity of obtaining computed answers to important problems required technical departments to improvise very special-purpose methods. One example is the work of Clarence A. Lovell and Linus E. Kittredge on traffic-congestion problems for the first crossbar switching system [1]. They managed to use punched-card equipment for much of their work, but to handle the crucial link-matching

---

This document is an OCR-generated rendition of Computing Science Technical Report No. 99, prepared in 1982, mostly by Brown and Holbrook. It served as a draft for a section of the series *A History of Engineering and Science in the Bell System*, in the volume subtitled *Communications Sciences* (AT&T Bell Laboratories, 1984: ISBN 0-932764-08-1) as Chapter 9: Computer Science, on pages 351-398. There, its principal authors are identified as W. S. Brown. B. D. Holbrook, and M. D. McIlroy. The content of the CSTR and the book rendition overlap considerably, but are not identical; the book's chapter is somewhat updated and edited, and includes more Unix-related material.

Editing errors introduced during fixup of the OCR are the responsibility of Dennis Ritchie.

phase of their job they had to build a large mechanism that included two or more Monroe desk calculators, some moving belts whose motion was tied to the calculators, and a number of clerks to move counters onto and off the belts and to transfer numbers between counters and calculators. With this analog mechanism for a basically digital problem, Lovell and Kittredge provided the information required to engineer early crossbar systems.

The probability studies needed for initial engineering of the early multichannel telephone-transmission systems provide a second example of special-purpose methods improvised to obtain computed answers. The distribution of instantaneous voltages in the speech of individual channels was known, mainly by the use of sampling equipment developed by Hugh K. Dunn of the Bell Labs Acoustic Research Department. To obtain the distribution for multichannel speech as a function of the number of channels, Bernard D. Holbrook recorded telephone speech on high-quality phonograph records, and used simple electrical analog adders to combine the output of four such records and rerecord this on a "four-voice" record [2]. By repeating this process to obtain sufficient samples, and using the original sampling equipment to measure the distributions for various numbers of speakers. he made it possible to design economical multichannel amplifiers with adequate load-carrying capacity. His procedure, of course, amounted to the use of an analog computer, built of necessity out of components that were then readily available.

3. TECHNICAL BACKGROUND FOR COMPUTERS

The first specific suggestion for doing arithmetic by electrical methods came from Sumner B. Wright and Edmund R. Taylor [3] of the Development and Research Department of AT&T [4]. They were not at all concerned with computational problems, but rather with the mechanization of the control of transatlantic radiotelephone facilities. Here it was necessary to adjust the gain of certain sections of the transmission paths to insure that the actual radio links were used at their maximum capacity, but without permitting overloading if either the speech volume or the noise level changed substantially; heretofore this had been done manually by technical operators observing suitable meters. Wright and Taylor's mechanism was basically an analog adder that used the algebraic sum of the rms values of several rapidly varying waveforms to effect the necessary control. It took some time for this idea to be widely used, and then it took rather a different form from the initial proposal; the delay was essentially because the invention was a bit ahead of the state of the art.

Fortunately, the state of the art was rapidly improving. On the analog side, Harold S. Black's invention of the feedback amplifier in 1927 and Hendrik W. Bode's development of mathematical methods for designing it to specified tolerances led to the precise, stable, reliable vacuum-tube circuitry that made the amplifier a precision component for an accurate computer [5]. These developments also permitted the development of servomechanisms of comparable accuracy. On the digital side, the pertinent history goes back to 1906 when Edward C. Molina's invention of the relay translator triggered the developments that ultimately resulted in the panel dial system [6]. During this period of development, engineers learned how to use relays to handle all kinds of duties that had previously required the attention of an operator, and by about 1930 the design of relay circuits was a sophisticated art. It was, however, an art difficult to teach to novices. But in 1937, Claude E. Shannon showed how to use Boolean algebra for the synthesis, analysis, and optimization of relay circuits, and the design of relay circuits became no longer a somewhat esoteric art, but a science that could be taught as a straightforward engineering discipline [7].

4. ELECTRICAL ANALOG COMPUTERS

Since useful analog computers could be built without modern electric technology, they were in constant, though limited, use long before the digital computer. In many fields they were very valuable, as for instance where mechanical models of continuously changing problems could be set up on a machine in scale-model form. Jacob Amsler's planimeter and Lord Kelvin's ball-and-disk integrator are early examples. Amsler's polar planimeter, invented about 1854, could readily measure the area of any plane shape. The operator simply traced the outline of the shape with a pointer attached to the mechanism. The difference of the readings on a graduated roller taken before and after the trace was the area of the shape [8]. Kelvin's integrator was the heart of what was sometimes called "the great brass brain," a machine that predicted the tides for any port for which the tidal constituents had been found -- not merely the times and heights of high water, but also the depth of water at any and every instant for a year or more in advance [9].

During the early 1930s, Vannevar Bush at the Massachusett Institute of Technology greatly increased the flexibility of the analog computer by applying electrical control and drive equipment; the computation itself was still based on an improved mechanical ball-and-disk integrator. At about the same time, comparable mechanical analog computers, with electrical follow-up servos, were beginning to be used by the United States Army and Navy -- particularly the latter. These computers notably improved the performance of their medium and heavy guns.

Bell Labs made some use of the Bush equipment, and also built some small analog computers for special purposes. One example was the Isograph, a mechanical, two-dimensional analog of the one-dimensional harmonic synthesizer built around the turn of the century by Albert A. Michelson and Samuel W. Stratton [10]. It was designed to find the complex roots of polynomials, a necessary step in the design of many types of filters and networks. The Isograph did its job, but not well enough to compete successfully with the desk calculator. During World War II, it was given to Princeton University for instructional use but fell a victim of wartime scheduling difficulties: it was shipped by rail to Princeton and left overnight on a railway platform without a protective cover. During the night there was a heavy rain, and the resulting rust made the isograph no longer a precision instrument.

The pressing need for better control of antiaircraft guns led - just before this country entered World War II - to the development by Bell Labs of an electrical analog computer, first conceived in 1940 by David B. Parkinson and Lovell. This computer used shaped wire-wound potentiometers and precision vacuum-tube amplifiers to perform standard arithmetical operations, and led directly to the M-9 gun director, which became the Army's mainstay for fire control of heavy antiaircraft guns [11]. The first production M-9 was delivered to the Army on December 23, 1942, and others followed very rapidly. These gun directors did yeoman service on many fronts; their finest achievements were against the German V-1 buzz bombs during the Second Battle of Britain. During the month of August 1944, over 90 percent of the buzz bombs aimed at London were shot down over the cliffs of Dover; in a single week in August, the Germans launched 91 V-1's from the Antwerp area, and heavy guns controlled by M-9's destroyed 89 of them.

A number of other fire-control computers for antiaircraft guns and one for control of coast-defense artillery were built during the war. While none of these computers was placed in regular operation, their development led to further advances in the technology of electrical ana log computers. A more detailed account of these developments is given in Chapter 3 of the second volume in this series, *National Service in War and Peace, 1925-1975.*

All of these military analog computers were designed to perform elaborate, but very specific, computing tasks. After the war, a need was soon felt for computers that could solve a variety of mathematical problems, particularly those beyond the grasp of the first relay computers. To find a way of solving the growing number of problems not amenable to other methods of computation, Bell Labs -- like other members of the technical community -- soon turned to the computer. In addition to relay computers (discussed below), Bell Labs developed a general-purpose analog computer (GPAC) [12]. Nicknamed Gypsy, the computer was designed by Emory Lakatos of the Mathematical Research Department. In its construction, a good many left-over parts from uncompleted wartime computers were used.

Like the military analog computers, this machine used electronic circuits to perform addition, subtraction, multiplication, division, integration, and differentiation. Unlike the military computers, its circuit configurations were readily changed from problem to problem, which made it much more flexible to use. It also used, as its normal output mechanism, precise electrically driven plotting boards developed in connection with wartime gun-director work. Although its accuracy was only in the range of 0.1 to 1 percent, this was adequate for many engineering applications, especially since some of the problems that Gypsy was able to solve, such as nonlinear differential equations for relay design, were otherwise so extremely laborious to handle that without such a computer only very rough approximations were available. The first Gypsy was placed in service in 1949 (Fig. 1) and proved so useful that a duplicate was built a few years later. The two machines were arranged so that for small problems they could be used independently, but could be coupled together when large problems had to be solved. They were, however, replaced in 1960 by a commercially built machine that, comprising ten years of new developments, was faster, more compact, and much quicker in changing over from one problem to another. The Gypsies were then given to the Polytechnic Institute of Brooklyn for educational use.

5. DIGITAL COMPUTERS

5.1 The Complex Number Computer

In 1937, George R. Stibitz, a Bell Labs research mathematician, was well aware of the growing need for improvements in numerical computation and also of the logical capabilities of relay circuits. Since he saw both the need and a practicable means of satisfying it, he proceeded to sketch out a preliminary design for a relay calculator. His initial plan consisted of a machine that worked internally in the binary system, with decimal input either from a keyboard or teletypewriter paper tape, and decimal output either on paper tape or teleprinter. Relay circuitry would take care of binary-decimal conversion in either direction. His plan also provided for internal memory (relay registers) and for TELETYPE® tape facilities to handle programs and subroutines and to provide additional external memory. The machine would be constructed from existing telephone components: relays, sequence switches, and standard Teletype equipment. A careful examination of the possible uses of such a machine resulted in a decision to build first a smaller and simpler machine that would try out most of the essential features; the resulting experience would be of great value in design of a second and more elaborate machine.

At the time that Stibitz was working on his computer, there was a great need for improvements in means for accurately performing standard arithmetic operations on complex numbers. There were three computing groups at Bell Labs who were spending a large proportion of their time doing such calculations on desk calculators, a job that could be handled by a relatively simple machine of the type envisioned by Stibitz. This machine was designed by Stibitz, and engineered and constructed during 1938 and 1939 under the direction of Samuel B. Williams, an experienced relay-system design engineer [13]. Because of the time needed for relay circuitry to do extensive binary-decimal conversion for input and output, Stibitz revised his initial proposal in favor of operating throughout on a binary-coded decimal basis, using four relays per decimal digit, with astutely modified binary coding *within* each digit.

The computer consisted of a standard relay rack, on which were mounted 450 relays and ten crossbar switches (Fig. 2). There were two separate calculator units, one to handle the real parts of complex numbers, the other for the imaginary parts. Input and output could handle numbers of up to eight decimal digits, with two extra internal digits to minimize round-off errors. The computer itself was locked up in a large closet, which was opened only for maintenance. Its users were provided with three operator stations, each with a keyboard for input and a standard teletypewriter for output (Fig. 3). The keyboard also made it possible to choose the complex-number arithmetical operations to be performed. The multiplication and division keys each directed a complex numerical operation by calling an appropriate subroutine of about a dozen steps. This effected the required complex operation by two calculator units, each working only on real numbers. Three operator stations were installed on different floors of the Bell Labs building on West Street in New York City. Each was close to one of the three groups expected to make the most use of the computer. This was the first instance of either remote or multistation computer-terminal facilities, although, of course, the limited speed of the relays in the computer permitted only one operator station to be used at a time.

The machine was completed in October 1939, and, after thorough testing for performance in actual operation, it was placed in routine service on January 8, 1940. It remained in service until 1949, continuously performing accurate and rapid calculations. During World War II, the great increase in the work load of the network design groups, its principal users, kept it almost continuously busy from 8:00 a.m. to 9:00 p.m. six days a week. Since the machine had been built as a demonstration model before the war, it was not equipped with many of the self-checking and contact-protection facilities that were standard in dial-control central offices; and the war prevented design and construction of the second and more elaborate machine that was initially envisioned. As a result, it became necessary, late in the war, to take it out of service for two days, while special maintenance tools (developed by Western Electric for the relief of central offices with similar complaints) were used to strip the badly-worn contacts from the computer's relays and replace them with new metal.

Thus, Stibitz' original Complex Number Computer, later known as the Model I, remained in service for over nine years, until replaced by the Model VI. It was the first electric computer to perform its arithmetical operations in basically binary fashion, the first to be placed in routine operation for general use, and the first with either remote or multistation terminal facilities.

The first public demonstration of the complex computer took place on September 11, 1940, before a meeting of the American Mathematical Society at Hanover, New Hampshire. One of the operator consoles from the West Street building, modified to communicate with the computer over a standard long-distance teletypewriter circuit instead of the multiconductor cable used locally, was installed in the lecture room at Hanover, and members of the audience were invited to use the keyboard to give the computer problems involving addition, subtraction, multiplication, or division of complex numbers [14]. Among the interested participants was Norbert Wiener who was an M.I.T. mathematics professor at the time. The circuits transmitted the input to the computer's relay equipment in New York and the results back to the Hanover teletypewriter; the answers returned in less than a minute. This remote-control operation, not to be duplicated anywhere for ten years, foreshadowed the use of telephone and radio circuits for computer data transmission. This became commercially important in the mid-1960s and has since shown almost explosive growth. Another result of the Hanover demonstration was that mathematicians from many parts of the country began, for the first time, to think seriously about new methods in computation.

5.2 Relay Digital Computers in World War II

The successful development of electrical analog computers for gunfire-control purposes triggered a demand for a great deal of highly routine computation. Initially, this computation was used in the performance tests of gunfire-control equipment as it came off the production line, and later in the investigation of the effects of new enemy tactics on the behavior of available equipment and the value of possible design modifications as countermeasures. The required computing was almost always within the scope of desk calculators, but the load was immediately seen to be much greater than could be handled with available personnel, equipment, and methods of system organization. The digital techniques provided by Stibitz and Williams were therefore applied, and as a result, Bell Labs developed three additional relay computers during the war [15]. These were designed as special-purpose machines to meet very specific needs, but turned out to be sufficiently flexible to handle many other types of problems. These machines are described in more detail in Chapter 3 of the second volume in this series, *National Service in War and Peace, 1925-1975.*

All three of these computers used punched paper tape for data input and output, and also for program input. Frequently used subroutines were punched on looped tapes so that they could be called from the main program as needed. The Model II relay computer (Fig. 4) contained 440 relays and five pieces of TELETYPE equipment. It was designed to perform linear iterative operations on numbers obtained from an input data tape. Its repertoire of arithmetic skills was thus very limited. The Model III and Model IV relay machines were designed for Army and Navy use, respectively, and were much larger and more powerful than the Model II (Figs. 5 and 6). Each contained about 1400 relays, 10 storage registers, and 7 pieces of TELETYPE equipment. All three machines had the standard dial-system features needed for reliability and maintainability.

The Model II machine was placed in service in September 1943, the Model III in June 1944, and the Model IV in March 1945. All of them operated regularly seven days and seven nights a week, usually unattended, and together they did the work of at least 100 desk calculators. All of them were later modified to extend their capabilities, and they remained in service for 13 to 15 years after the war--several years after much faster commercial electronic computers were readily available.

5.3 Model V Relay Computers

In 1946, Bell Labs made a significant contribution to the evolution of modern computers with the delivery of a Model V relay computer (Fig. 7) to the National Advisory Committee on Aeronautics (NACA) at Langley Field, Virginia [16]. In the following year, a duplicate Model V was delivered to the Army's Ballistic Research Laboratory at Aberdeen, Maryland. These computers easily represented, so far as size and flexibility were concerned, Bell Labs most ambitious computer development project until then. They were specifically designed to be general-purpose computers. Each used about 9000 relays and had two separate processors; the system design permitted a maximum of six processors per machine. One of the machines had three problem positions installed; the other had four. While a machine was in continuous operation, a new problem could be loaded on an unused position and be automatically picked up when a processor was free to handle it. Each of the problem positions had a tape reader for input data, as many as five readers for programs, and up to six readers for tabular data. As in the wartime machines, subroutines were punched on looped tape so that they could be repeatedly called from the main program as needed, and the tape devices for intermediate or tabular data were arranged to permit both forward and backward searching to find required locations in storage as rapidly as possible. Such searching could go on independently of calculation.

Since processing was handled on telephone relays, with operating times measured in milliseconds, there was excellent speed-matching between internal operations, storage, and input than the Model I. Essentially, the Model VI was a somewhat simplified version of the Model V, since it had only a single processor and a smaller number of problem positions. However, it had new and interesting features of its own, notably fast internal storage for several hundred semipermanent subroutines. This was provided by a "Dimond-ring translator," invented by Thomas L. Dimond, and used in the No. 5 crossbar dial system to provide rapid conversion from the code describing the main-frame location of a calling line to the caller's number as listed in the telephone directory [18]. This translator consisted of about 80 air-core coils, each of which would trigger an associated gas tube when a suitable pulsed current flowed in any one of the wires threading that particular coil. In the computer application, a subroutine was "programmed in" by threading a wire loosely from a numbered pulse terminal through a correctly chosen subset of the available coils to a common return terminal. This library of subroutines operated at six levels of precedence. The highest level was called by an order punched on a program tape, and each level could call in sequence several subroutines at lower levels. The bottom level, of course, consisted of the basic instructions built into the hardware of the machine. These extremely flexible subprogramming facilities avoided the cumbersome tape-handling required in the earlier Bell Labs relay computers, and thus made program preparation for the Model VI a great deal easier.

The Model VI also had a "second-trial" feature, which operated automatically when the control circuit failed to receive the usual signal indicating satisfactory execution of the instruction or operation called for. Experience on previous relay computers had shown that a sizeable proportion of machine stoppages resulting from relay-contact troubles would clear themselves when the relay at fault operated again. This automatic second trial proved effective in permitting much longer periods of uninterrupted computer operation. In addition, if the machine was operating unattended and the second trial failed to check, the problem was abandoned and the master tape searched for the beginning of the next problem, which was then loaded. When work was available, it was customary to load the machine late every afternoon with enough problems to keep it busy until morning; on Friday afternoons enough could be loaded to occupy it until Monday morning. In these circumstances, the machine was started, the room lights turned off, and the door Locked. Sometimes one or two of the problems would be found abandoned in the morning, but they could then be rerun (perhaps after some maintenance). Models V and VI represented the high point of the relay-computer art: their successors almost all used electronic rather than electromagnetic apparatus to permit higher speed. The Bell Labs machines were equipped with very dependable heavy-duty relays perfected for telephone-switching applications; the use of such relays, together with the provision of extensive self-checking features, resulted in high availability and high accuracy. In fact, during their entire working lives, only two errors due to machine failures were reported from all three Model V and VI machines. Their operating times were, however, quite slow-it took about a second to perform a multiplication, 2.7 seconds for division, 4.5 seconds for a square root, and as much as 15 seconds to calculate a logarithm. But reliability and accuracy were the main objectives, not speed. And the reliability of these plodding, meticulous computers was truly remarkable.

The Model VI remained in service at Bell Labs Murray Hill location until 1956, when it was replaced by a much faster commercial electronic computer. It was then given to the Polytechnic Institute of Brooklyn, where it was used for both instructional and research purposes, and operated reliably and with negligible difficulty until March 1960, when it was again replaced by an electronic computer. It was then offered to several smaller colleges, and finally given to the University of Bihar in India.

5.5 Relay Computers for Message Accounting

The successful and reliable operation of the Model I through Model V relay computers was influential in determining the course of development of the accounting center equipment for the automatic message accounting system. This system mechanized most of the operations involved in billing telephone subscribers for their detail-billed or bulk-billed calls. The information needed for billing such calls was collected automatically (initially only in the local dial central offices; after about 1953, also in dial tandem or dial toll offices). Then, from this data, the billing was assigned to the correct customer, printed out, and delivered. The first such accounting center was opened in 1948, coincident with cutover of the first No. 5 crossbar office, which was equipped for this type of operation [19]. Early accounting centers depended heavily on repunching paper tape for much of their operation, but relay computers were used for doing the necessary arithmetic; later, improved assembler-computers were installed that used relay-computer technology for sorting calls to customer's numbers as well as for the arithmetic needed for determining charges. About a hundred of the combined assembler-computers were built, and they, together with their simpler predecessors, provided reliable and accurate billing facilities until the relay-computer accounting center facilities were gradually replaced (mostly in the 1960s) by electronic computers of standard commercial types. These accounting centers represented by far the major application of relay-computer techniques in this country.

To wrap up the story of relay digital computing techniques, we note that the type of traffic problem investigated for the No. 1 crossbar system by Lovell and Kittredge in the mid-1930s was, 15 years later, handled for No. 5 crossbar by George R. Frost, William Keister, and Alistair E. Ritchie, who built for this purpose a very specialized relay computer [20]. This essentially used relay circuitry to do what Lovell and Kittredge had done, much more clumsily, by punched-card methods; but it is interesting to note that it was still necessary to use human operators to handle the link-matching job.

5.6 Other Early Electrical Digital Computer Developments

Pioneering work in the computer field in the late 1930s and early 1940s was by no means limited to Bell Labs. The first proposal that actually led to construction of an electrical digital computer was that of Konrad Zuse in Germany, who in 1936 applied for a German patent on a binary computer [21]. The patent states that the machine could be constructed either from relays or from "mechanical coupling and uncoupling devices" with equivalent logical results. Apparently because of the cost of relays, he built his first machine almost entirely of such mechanical switching elements. These consisted of plates movable at right angles to each other and constrained by attached pins working in slots to cause or prevent transmission of "yes-no" values from plate to plate. This mechanical binary computer (Z1) was completed in 1938, but the mechanical switching elements proved unsuitable for carrying out arithmetic, and the machine's operation was unreliable. Zuse then built a small experimental arithmetic unit (Z2), which was coupled with the mechanical memory of Z1. With this unit, some simple formulas could be calculated, but practical utilization of the machine was still not possible. This was followed by the Z3 machine, built entirely of telephone relays, about 2600 of them. This machine was begun in 1939 and placed in service in 1941, at least a year after Bell Labs Model I computer began routine operations. The Z3 and its predecessors were destroyed in air raids during the war. Zuse's work was unknown outside Germany (and probably little known in that country) until after the end of the war.

In the meantime, the Harvard University Mark I computer was being developed and constructed in Cambridge from 1938 to 1945 under the direction of Howard H. Aiken [22]. This was an extremely large electromechanical computer, more than 50 feet long and containing over 750,000 parts. Its arithmetic operations were, however, done on ten-position rotary counters adapted from IBM punched-card equipment and rather like the step-by-step switches used in many dial telephone systems. Enough of these were provided to handle fixed-point numbers of 23 decimal digits. The sequencing of operations, under control of programs punched on paper tape, was handled by relay circuitry. Like all electromechanical computers, it was slow - it took about 4 seconds for multiplication, 16 seconds for division. There were no features for checking reliability-these had to be programmed. Slow and cumbersome as it was, this giant calculator was the first general-purpose machine designed to carry out enormous arithmetical jobs. It was specifically designed for the construction of extensive tables of mathematical functions, and in this characteristic was unlike the Bell Labs machines. The Mark I computer was placed in operation at Harvard in 1945, and was followed by the Mark II, also designed by Aiken [23]. The Mark II used relays for calculation as well as control, and was comparable to the Bell Labs Model V machines in size and capabilities. It was installed at the Naval Proving Ground at Dahlgren, Virginia.

The most significant achievement of this period was the development of the ENIAC computer at the University of Pennsylvania's Moore School of Engineering under the direction of John W. Mauchly and J.

Presper Eckert [24]. For the first time, the high-speed capabilities of vacuum-tube operation were utilized, permitting speeds that almost immediately made relay computers obsolete. When ENIAC (an acronym for Electronic Numerical Integrator And Computer) was completed in April 1946, it contained about 18,000 vacuum tubes and a battery of fans and blowers to keep the internal temperature below the point where it would cause damage. There were, however, no air-conditioning facilities, and repair was a constant problem. But ENIAC's electronic operation permitted a very impressive step-up in speed: it could do 5000 additions a second, though its speeds for multiplication, division, and reading numbers from punched cards were a good deal slower. ENIAC, like the Mark 1, was a decimal rather than a binary machine; it is fair to say that it used vacuum tubes to simulate, at electronic speed, the operation of the rotary counters of the Mark I. The machine had no built-in checking systems, and its storage capacity was quite limited, since it used expensive vacuum-tube counters for this purpose. Its high internal speed prevented the use of the inexpensive paper tape and cards that so well matched the internal speeds of the Bell Labs and Harvard relay machines. Like the Mark I, ENIAC was intended basically for calculating large tables of mathematical functions, and its programs could only be changed by a complicated process of altering plugboards and setting many switches. Despite these difficulties ENIAC, rather than the electromechanical computers, pointed the way to the future. The vacuum-tube machines that followed were great improvements, but the most dramatic new achievements had to await the advent of the transistor.

Another important step in the history of the digital computer was John von Neumann's conception (first described in 1946) of the general-purpose computer with storage facilities shared by programs and data. The first use of this idea was in ENIAC itself, which was extensively modified to incorporate this new concept. The EDVAC, built at the Moore School and placed in service in 1950, was the first computer designed from scratch as a stored-program machine. The flexibility of operation thus obtained was the key that made future electronic computers so easily applicable in a wide range of problem areas. The importance of the concept depended on providing a storage organization, properly matched in speed to the calculating capabilities of the machine. The methods used in the larger Bell Labs computers, particularly in the Model VI, were entirely adequate for relay arithmetic, but it took some time to develop a corresponding match for electronic calculating speeds.

At the end of World War II, Bell Labs management planned a program of development work required to provide urgently needed, new Bell System telephone facilities, which had been delayed for five years while 80 percent of Bell Labs staff was devoted to the country's military wartime.-needs. One possibility considered was work on vacuum-tube computers, since this was by then seen to be the way to the future. The importance of the computer art was clear, but it was also clear that the kind of people who could contribute significantly in a new field, just beginning to be explored, were more urgently needed for long-postponed telephone development work. As a result, there was a hiatus in Bell Labs computer activity between the relay era and that of the transistor.

6. SOLID-STATE TECHNOLOGY AND THE TRANSISTOR

The great size and heavy power drain of vacuum-tube digital computers like ENIAC and its immediate successors could have severely limited their growth in complexity and efficiency. As it happened, the expanding computer art paralleled an equally dramatic growth in solid-state technology. This trend first became evident in the growing use of passive devices for doing many of the necessary internal operations in a computer: bistable magnetic cores for fast, compact, relatively cheap, random-access memory, and crystal diodes for handling most of the detailed logical operations needed in calculation and control. Two of the most-used diode logic circuits, the AND gate [25] and the OR gate [26], had been invented about a decade earlier at Bell Labs in connection with exploratory work on new dial-switching techniques. The magnetic core and the diode still required use of vacuum-tube pulse amplifiers to restore signal levels, but the total number of vacuum tubes was greatly reduced, together with the power requirements and the physical size of the tubes. As a result, computers became smaller while their performance became substantially better. The commercial computers of the latter 1950s were typically based on this use of solid-state logic with vacuum-tube amplification.

In late 1947, several years before magnetic cores and crystal diodes began to be used extensively in computers, John Bardeen, Walter H. Brattain, and William Shockley at Bell Labs discovered the transistor effect. Just as the new vacuum-tube technology had ended the day of the relay computer, this discovery foreshadowed the end of the vacuum tube in digital computers. The new technology took over a decade to come to fruition -it was first necessary to learn how to manufacture transistors in adequate quantities and to suitable specifications. Nevertheless, the transistor made possible the all-solid-state computers of the 1960s.

Probably the first computer-like transistor circuits in regular operation were those in a transistor gating matrix built by Walter H. MacWilliams, Jr. in 1949 as a small part of a "simulated warfare" computer [27]. Two general-purpose, all-solid-state digital computers, TRADIC (TRAnsistor DIgital Computer) and Leprechaun, and a large special-purpose machine for a Naval gunfire-control system were developed by Bell Labs between 1952 and 1959. These and other defense-related computer projects are described in Chapters 10, 11, and 13 of the second volume in this series, *National Service in War and Peace, 1925-1975.*

In recent years, the Bell System and the world of computers have had an increasingly close relationship. Telephone facilities are being used more and more to transmit data to, from, and among computers, and the Bell System makes more and more use of computers in its day-to-day operations. Similarly, the use of computers in Bell Labs research and development work and of computer-born technology in both transmission and switching applications in the Bell System have grown increasingly in importance.

In the early 1950s, Bell Labs problems occasionally became large enough to require the use of machines of greater size and power than the relay computers. Time was therefore rented, as needed, on the IBM 701 and the Univac. The load of smaller problems also increased, and in 1952 Bell Labs acquired an IBM Card Programmed Calculator; this was replaced in 1955 by an IBM 650 machine, and a second 650 was installed a year or so later as the computing load continued to grow.

7. EARLY PROGRAMMING LANGUAGES

One of the effects of this load growth was to present Bell Labs with its first real software problems. The Model VI relay computer was, for its time, fairly easy to program. Use of the library of stored subroutines avoided much detailed and repetitive programming. Most of the programming was done by people skilled in using the machine, and there were enough of these people to handle the load. By the time the IBM 650 arrived, the situation had begun to change. More and more scientists and engineers had useful jobs for the computer, and increasingly they wanted to handle their jobs experimentally -- that is, they wished first to calculate what would happen if they did things in the standard way, and then, after looking at the initial results, see what would happen if they changed design of the circuit or mechanism they were concerned with in two or three ways suggested by the first attempt.

To make this kind of operation really practicable, Bell Labs developed new problem-oriented programming languages that permitted such users to make effective use of the machine without the necessity of becoming completely familiar with programming in the machine’s "native" language. These languages made floating-point operation available to the user (although the machines themselves operated in fixed-point arithmetic), greatly simplified the addressing of data in the memory, and provided useful diagnostic information as to program malfunctions. There were two such languages, each with specific advantages for certain types of work: the L1 language [28], developed by V. Michael Wolontis and Dolores C. Leagus, and the L2 language, developed by Richard W. Hamming and Ruth A. Weiss. They proved very convenient in operation, and both of them were released to users outside of Bell Labs, who usually referred to them as Bell 1 and Bell 2. In the late 1950s, at least half the IBM 650s doing scientific and engineering work used either Bell 1 or Bell 2. One organization became so fond of Bell 1 that, when its 650 was replaced by the more powerful IBM 1401 (which came complete with excellent IBM problem-oriented software), they went to the trouble of writing their own Bell 1 interpreter for the new machine.

With this software, the IBM 650s served Bell Labs scientists and engineers very well for several years. The operating procedures were straightforward: the user’s program and data were keypunched and proofread, then the card deck, preceded by the L1 or L2 interpreter, was fed into the IBM 650, and the output appeared at the other end of the machine, also punched into cards. The output deck was then printed for the user on an IBM tabulator. If the user feared there might be undetected errors in the program, it could be run in tracing mode to obtain a complete listing of executed instructions. Clean decks were run by an operator without the user being present. During the last year of use of the 650s, the machines ran pretty well around the clock; on each of the second and third shifts, one operator ran both machines with no trouble.

After a short period of instruction and practice, most scientists and engineers did their own programming, with occasional help from a few skilled mathematician-programmers who were available for consultation. Some special jobs had to be programmed in machine language, but in general the operation was largely "open shop" (programmed by users) rather than "closed shop" (programmed by professionals). On the whole, the users preferred it that way. They usually got answers faster, they knew what was going on, and they had no worries as to whether a programmer quite unfamiliar with their special field really understood the problem for which an answer was sought.

By 1957, the computing load at Bell Labs was straining the capacities of the IBM 650 machines, even when operated on a full three-shift basis. This applied not only to the total load but also to the increasing size of individual jobs. Accordingly, arrangements were made to replace the 650s by the much larger and faster IBM 704. For this, IBM provided a problem-oriented language called FORTRAN (FORmula TRANslation), which replaced L1 and L2 to considerable advantage. There was also a symbolic assembly language (SAP), which greatly reduced the burdens of machine-language programming for problems beyond the capabilities of FORTRAN. SAP was very useful, since the number of such problems grew when users became more aware of the logical powers of electronic computers.

8. OPERATING SYSTEMS

Full use of the much greater speed of the 704 required the central calculator (the main frame) to operate from magnetic-tape input to magnetic-tape output. Punched-card input was handled by off-line card-to-tape converters. The output tape could be printed by tape-driven printers or, if necessary, reduced to card format by tape-to-card punches. In addition, extra on-line tape units could be used to read data tapes or program tapes, to tape output material needed for later jobs, or to serve as auxiliary external storage for very large jobs.

The great increase in operating speed demanded software that was designed to use the machine itself to do many of the things that operators did in the days of relay machines and the IBM 650s. Such programs are called monitors or operating systems, and Bell Labs pioneered in their early development. These are large and complex packages of software. Their first function is to act as automatic operators, as, for instance, in transferring from one job to the next far faster than can be done by a human operator; there must still be an operator, but only for doing things beyond the machine's capability, such as mounting and dismounting magnetic tapes, or for taking care of situations requiring human judgment rather than routine response. They also provide computer users with ready access to standard compilers and assemblers (such as those for FORTRAN and SAP), to standard input-output routines, to libraries of previously developed routines for purposes such as calculating standard mathematical functions, and to flexible diagnostic facilities that permit program debugging and testing to be done off-line. They thus permit maximum use of the expensive central computer and at the same time substantially simplify the programming of engineering and scientific work.

The first Bell Labs operating system, BESYS-2, was written for the IBM 704 by George H. Mealy and Gwen J. Hansen, beginning in mid-1957. It was developed because, although some more primitive monitors were then available, there were none at Bell Labs. When the IBM 704 was placed in operation at the Murray Hill laboratory in April 1958, it was under control of the BESYS-2 operating system, as was the additional 704 installed at the Whippany laboratory at the end of 1959. This same basic system, updated from time to time as needed, was used on subsequent IBM 700/7000 class equipment used at Bell Labs. It was also used on many other installations of similar IBM equipment, where it was obtained either from Bell Labs or through the IBM SHARE organization of users of such equipment. It also had a wide impact on manufacturer-provided software; several later operating systems were based at least in part upon it.

Over the next decade, this basic operating system was repeatedly modified to handle changes in computer hardware or to provide additional desirable programming or operating facilities. Some of the major changes are listed in Table 1.

In preparation for the advent of the next generation of computers, it was necessary to provide for operation--on the successor machines--of programs written for, and often heavily used on, IBM 7000-class machines. This permitted an extended period for program conversion, at a relatively modest cost in extra machine time. For this purpose, Ronald E. Drummond, Hansen, and Frederick T. Grampp developed a

Table l. Chronology of the BESYS Operating Systems

System	Modification	Service Date
BESYS-2	This was the basic operating system to which the BESYS 3, 4, 5, and 7 modifications were made. It controlled an IBM 704 machine	April 1958
BESYS-3	Controlled the operation of the IBM 7090 machine.	July 1960
BESYS-4	Included first input-output system with full automatic blocking and buffering* and unique computer-controlled tape switching equipment.	April 1962
BESYS-5	Modified to enable it to reside on disk file.	May 1963
BESYS-7	Included early version of a user-file system.!	May 1964

*Developed by Ronald E. Drummond.
Developed by George L. Baldwin and Henry S. McDonald.
! Developed by Ronald E. Drummond.

"7094 emulator," called BE90, for use on the IBM 360/65. This system permitted running a program designed for and operable on a designated source machine and operating system (in this case, the IBM 7094 operating under BESYS-7) on the target machine (the IBM 360/65). It handled the entire job: operating system commands as well as the user's program for the specific problem to be done. It was installed at the Holmdel and Indian Hill laboratories in March 1968, and emulated both BESYS-7 and IBM's IBSYS operating system until early 1972, almost three years after the departure of the last Bell Labs 7094 in March 1969.

Meanwhile, an early time-sharing system called CTSS (Compatible Time Sharing Service) was developed on the IBM 7094 at the Massachusetts Institute of Technology [29] . Then, in 1964, M.I.T. joined forces with Bell Labs and General Electric for the research phase of an ambitious successor system called MULTICS (MULTiplexed Information and Computing Service) to provide access to a central GE 645 computer and its file system for a large community of users at separate remote consoles [30]. At about this same time, the Bell Labs organization developing electronic switching systems began preparations to use a similar IBM system, called TSS (Time Sharing System), on a duplex IBM 360/67 computer, which was to be delivered to the new Indian Hill, Illinois, laboratory in June 1967. Together with other early key customers, Bell Labs significantly influenced the design and development of both TSS and the 67, which were essentially complete and fully operational in January 1970.

Even while work was proceeding on MULTICS and TSS, it became increasingly apparent that no single central computer complex could meet all the computing requirements of a large research and development organization. Accordingly, Bell Labs researchers pioneered in the use of relatively inexpensive minicomputers in the laboratory to permit scientists to interact with experiments in process in fields ranging from particle physics to human speech [31].

With the rise of minicomputers, computer scientists became intensely interested in small, simple, elegant, time-sharing systems. An outstanding example is the UNIX operating system developed in 1969 by Kenneth Thompson of Bell Labs for the Digital Equipment Corporation PDP-7, and later upgraded to run on the PDP-11 [32]. Among the novel features of the UNIX system are (1) its device-independent input-output system, which permits the user to direct output from any program to any suitable device or to a "pipe", which may then serve as input to another program, and (2) its elegant file system, designed by Thompson, Dennis M. Ritchie, and Rudd H. Canaday, which treats all files alike regardless of their form or content. By June 1976, the UNIX system was in regular use in more than 30 Bell Labs development groups supporting numerous other Bell System installations, and had been made available for educational and academic purposes to more than 80 universities.

9. HIGHER-LEVEL LANGUAGES

In the late 1950s. several symbolic assembly languages, such as SAP, had become available, and by 1957 some of these, including IBM's SCAT and SAP for the 704 machine, permitted users to define macroinstructions (often called macros) as shorthand for frequently occurring sequences of machine instructions. Then in 1959, M. Douglas McIlroy and Douglas E. Eastwood of Bell Labs introduced conditional and recursive macros into SAP, and in 1960 described how macros could be used to extend any programming language to meet the user's own special requirements [33].

From the time of their introduction, the development of macro techniques has been vigorously pursued at Bell Labs, especially by Nicholas A. Martellotto, Hans Oehring, and Marvin C. Paull in the Process III assembler for the No. 1 ESS machine [34], by Marshall E. Barton in the SWAP assembler for the ESS and Safeguard machines [35]; and by Bernard N. Dickman in the SWAP-based CENTRAN compiler for the Safeguard computer [36]. Other macro-based high-level languages created at Bell Labs include the BLODI language by John L. Kelly, Jr., Carol C. Lochbaum, and Victor A. Vyssotsky for simulating sampled-data systems from their BLOck DIagrams [37], the L6 language by Kenneth C. Knowlton for list processing [38]; the GRIN language created by Carl Christensen for programs to support GRaphical INTeraction [39]; and the MUSIC V language by Max V. Mathews for musical composition [40].

In the early 1960s, David J. Farber, Ralph E. Griswold, and Ivan P. Polonsky recognized the need for better facilities for manipulating strings of characters and developed the language called SNOBOL (StriNg Oriented symBOlic Language) [41]. With its novel approach to pattern matching, SNOBOL proved both useful and popular. Further work led eventually to the more sophisticated SNOBOL 4 language, which is widely used both at Bell Labs and elsewhere in fields ranging from document formatting to theorem proving [42].

The general availability of SNOBOL 4 is due in large measure to the portability of its processor, which is specified in terms of a carefully chosen set of macros. In view of the wide variety of computers at Bell Labs and throughout the Bell System, and the ever-growing investment in software for those machines, there is an urgent need to achieve greater software portability without increasing programming effort or sacrificing efficiency. In the early 1970s, the development of the ALTRAN language (see Section XII) marked a major advance toward this goal, achieved by writing the system in American National Standard FORTRAN supplemented by macros. The permitted subset of FORTRAN is called PFORT (for Portable FORTran); it was defined by Andrew D. Hall, and its rules, including those that apply to communication between subprograms, are enforced by a verifier developed by Barbara G. Ryder [44].

Another important Bell Labs language contribution is the general-purpose C language, developed by Dennis M. Ritchie in the early 1970s [45]. Almost all of the UNIX operating system and its associated utility and command programs (see Section 8) are written in C, which incorporates a flexible system of data types as well as the control constructs recommended by modern insights into the structure of programs. The efficiency and readability of C have contributed greatly to the success of the UNIX system and have led to the development of C compilers on the IBM 370 and Honeywell 6000 computers, thus permitting programs developed under the UNIX system to be made available at the major Bell Labs computation centers and elsewhere.

The field of language design has been very fertile. Other languages are discussed in the following subsections, and a great many more will undoubtedly appear in the years to come.

10. DATA TRANSMISSION BETWEEN COMPUTERS

As previously noted, the first relay digital computer was introduced to the scientific community over the first computer data link - a teletypewriter circuit with slightly modified terminal equipment. By the mid-1950s, it was apparent that the electronic computers that were becoming available would require much higher-speed data transmission than could be provided by standard TELETYPE equipment. Bell Labs accordingly demonstrated in 1956 the use of dialed-up telephone circuits to provide direct magnetic-tape to magnetic-tape transmission of digital data at a speed of 600 baud, or about 10 times that of teletypewriters. The data were protected by parity checks, and records showing parity errors were automatically retransmitted. This demonstration was not, in fact, hooked up to a computer. Since there was at that time no agreement as to computer magnetic-tape formats, an ad hoc arrangement of the magnetic tape was used, which was prepared and printed out, at much lower speed, on standard Flexowriter [46] equipment. The demonstration did, however, show that tape-to-tape transmission of digital data could be achieved-at speeds reasonably matched to the computers of the time-over normal long-distance telephone connections dialed at random.

In the early 1960s, transmission facilities similar to those used in the 1956 demonstration were used to provide several branch laboratories with entry to Bell Labs major computing centers in New Jersey. This enabled the branch labs to resolve problems beyond the capability of their own modest computing facilities. The transmission facilities normally used voice-grade telephone circuits, usually those provided for interlocation telephone traffic. The detailed arrangements depended on the specific equipment available at the remote location. In January 1962, an enlarged Holmdel, N.J., laboratory was opened -- Bell Labs' third major location in New Jersey - and engineering and development groups began to move in from Murray Hill and Whippany. Moving about 1000 engineers and scientists with all their laboratory equipment is a sizeable logistic operation. The move took about eight months to complete. In the meantime, it was essential to provide first-class service on a large-scale computer to many of the groups being moved, in spite of the fact that for much of 1962 the actual computing load at Holmdel would be well below that required to justify the cost of an adequate large-scale installation.

To handle this problem, a computing center was established at Holmdel and initially equipped only with off-line input-output equipment and magnetic tape units. This center was connected to the IBM 7090 at Murray Hill by a 40.8-kilobaud Telpak-A data link, over which both input and output were transmitted tape to tape. This was the first large-scale, general-computing service ever offered at a location that was remote from the main computer, and it provided excellent service to Holmdel personnel until September 1962, when the load had grown to a point justifying installation of an IBM 7090 at Holmdel.

During this period the TELSTAR" satellite was placed in orbit, and arrangements were made to transmit a Holmdel output tape made on the 7090 at Murray Hill to Holmdel via the satellite. This involved transmission via microwave circuits to Andover, Maine, thence via the TELSTAR satellite to a receiving antenna at Crawford Hill, New Jersey, and finally over a short microwave link to the Holmdel laboratory. Since the satellite would not be in an accessible orbital position during the normal first shift of computer operation, the Murray Hill computation center during the afternoon of August 8, 1962 saved a Holmdel output tape of suitable length. This tape contained the output of 10 or 12 jobs, and consisted of 2891 tape records, mostly of 996 alphanumeric characters each, although the last record of each job was shorter. To avoid complaints about delays from the 10 or 12 Holmdel users, this tape was transmitted immediately to Holmdel over the 40.8-kilobaud data link. About 5:45 PM, the tape was retransmitted with complete success via the TELSTAR satellite when it became available: the transmission indicated that no record required retransmission because of parity errors. This caused some concern to the operators at both ends, since on the Telpak facilities, records occasionally had to be retransmitted because of parity errors caused by noise or crosstalk. But when Holmdel printed out the TELSTAR tape, it agreed completely with the earlier copy made after transmission via the Telpak link.

Late in 1963, the Telpak-A facility was replaced by a 1.5-megabaud TI carrier link, and in the winter of 1966-1967, the only computing service at Murray Hill was provided over this line by a pair of IBM 7094s at Holmdel. These computers also served the Indian Hill laboratory over a Telpak-A data link, which was later also connected to the laboratory in Columbus, Ohio.

Such high-speed data links also proved invaluable in providing continuity of service during unexpected outages. In one such instance, a flood in the Whippany computing center - which had also been provided with a high-speed data link - required its machine to be taken out of service. While the machine was carefully dried out and tested for accuracy, the Whippany computing load was adequately handled on the Murray Hill and Holmdel equipment. Another use of data links was in load equalization. If one center was overloaded, jobs that would have been delayed if done at the point of origin could be transferred to another machine for faster execution. This was done not only when loads were approaching the maximum, but also to counteract load fluctuations. There was, for example, a considerable period when both the Holmdel and Murray Hill installations were, on the average, comfortably loaded. However, the Holmdel center had a pronounced peak in its load at the lunch hour, while Murray Hill's load showed a valley at that time of day. This situation had an architectural origin: for most people at Holmdel, the computing center was located on the way to the cafeteria, while at Murray Hill the computer was inconveniently located in relation to the cafeteria. Use of the data links thus resulted in considerably faster execution of short Holmdel jobs, with hardly any effect on the service to Murray Hill users. Such data links also made it possible for centralized graphical output facilities to provide rapid service to users at other major Bell Labs locations (see Section I1).

By the late 1960s, work was progressing on techniques for forming networks of cooperating computers. In 1968, Wayne D. Farmer and Edwin E. Newhall demonstrated an experimental loop system for interconnecting digital devices [47]. In 1970, John R. Pierce proposed a larger loop network for high-speed data communications, with users responsible for their own signaling and error handling [48]. At about the same time, Alexander G. Fraser proposed and started constructing the experimental, high-speed, packet-switched SPIDER network, in which a central minicomputer switch and intelligent data sets provide error-control and flow-control services for attached computers [49]. By June 1976, several minicomputers in the Bell Labs acoustics research group had been connected by a loop system following Pierce's ideas, and SPIDER had grown into an internal network supporting about a dozen mini- and midi-computers with various services, including a network file-storage facility, a network printer, and access to the Honeywell 6070 computer in the Murray Hill computation center. Research in progress should expand our ability to share network resources and lead to simpler techniques to permit cooperation between programs being executed in different machines.

In early 1972, Allan R. Breithaupt and Martellotto proposed the connecting together of the large IBM batch-processing systems at Holmdel, Whippany, and Indian Hill. Connections again were made via Telpak-A data links, and IBM's associated support processor (ASP) was expanded considerably to support ASP-to-ASP communications [50]. By June 1976, the resulting Bell Labs Interlocation Computing Network, which had become fully operational and generally available in 1974, included three centers and over a dozen satellite locations across the country. If desired, a user at one site could run a job at a second site and direct the output to a third site.

11. OTHER TYPES OF INFORMATION TRANSFER

Since computers are machines for storing, retrieving, and processing information, computing scientists have always been vitally concerned with the transfer of information-not only between computer and computer, but also between computers and people, and between people and people aided by computers. Although information transfer, viewed broadly, is the entire mission of the Bell System, we shall exclude telephony from our discussion and focus on the parts that belong properly to computer science.

For communication between computers and people, words and numbers may be sufficient, yet for many applications a graphical or pictorial representation may be much more informative. To provide this type of output, Bell Labs installed a Stromberg-Carlson 4020 microfilm printer at Murray Hill in 1961. This device, when fed digital information in suitable format from a computer output tape, converted the information into graph or chart form and recorded it photographically on microfilm. By using standard, rapid developing and printing equipment, the information was ordinarily delivered to the user as an 8- by 11-inch graph, chart, or picture. After the Holmdel laboratory began operation, it was provided with this service, without complete duplication of facilities, by use of the high-speed data links described in the last section. These links were used to send output from the Holmdel computer to Murray Hill, and to return the graphical output very rapidly to Holmdel with the aid of Xerox picture-transmission equipment.

Various researchers developed new graphical-output software to make these facilities readily available to users. This software included Clement F. Pease's microfilm package of basic utility subroutines and James F. Kaiser's TPLOT graph-drawing subroutine [51]. In the first large-scale application of these facilities, Walter L. Brown and John D. Gabbe generated several thousand plots (see Fig. 10) from hundreds of thousands of measurements of the earth's radiation belts made by the TELSTAR satellites [52].

The cheapness of film production on the Stromberg-Carlson recorder suggested the use of movies. Accordingly, Robert M. McClure made a classified movie of a cloud of incoming enemy missiles and decoys, and Joseph B. Kruskal made a movie to display the iterations of his algorithm for multidimensional scaling. Then Edward E. Zajac conveyed the results of his computer simulation of satellite motion as a movie of a gyrating and tumbling box (see Fig. 11) [53]. A. Michael Noll made a stereographic three-dimensional movie, and Frank W. Sinden illustrated the educational potential of computer movies in his article "Synthetic Cinematography." [54]. At about the same time, Knowlton introduced a special movie-making language called BEFLIX (see Fig. 12), with which several award-winning scientific and artistic films have since been produced [55].

The first on-line "intelligent terminal," a Packard Bell 250 computer, was connected to an IBM 7090 at Bell Labs in 1964. Cooperating software on the two machines, developed by Elliot N. Pinson, allowed a single high-priority user at the 250 to interact both graphically and acoustically with calculations being performed on the 7090. Concurrently, Henry S. McDonald, William H. Ninke, and Christensen developed an intelligent terminal called the GRAPHIC l (see Fig. l3), incorporating a DEC PDP-5 minicomputer to avoid overburdening the main machine [56]. Following this, Ninke, Christensen, and Pinson developed the more advanced GRAPHIC 2 (see Fig. 14) [57]. The GRAPHIC 1 and GRAPHIC 2 were milestones in the evolution of remote computer terminals and represented a remarkable advance over the simple teletypewriter used by Stibitz in 1940 to demonstrate his complex number computer (see Section 5.1). The GRAPHIC 2, now manufactured for Bell Labs and Western Electric by the Digital Equipment Corporation, is widely used for the computer-aided design of printed-wiring boards, logic schematic drawings, and office equipment layouts.

Computers are also playing a growing role in the transfer of information among people. For example, many Bell Labs papers are composed at a teletypewriter connected to a UNIX system on a PDP-II (see Section 8), formatted by Joseph F. Ossanna's TROFF language, and typeset by a computer-controlled phototypesetter. Furthermore, the computer may be used to merge programs and data from other files into the text of the paper, and the author may choose to make its structure and/or, content dependent on the results of computations.

To help get Bell Labs papers promptly to the employees who need them, W. Stanley Brown and Joseph F. Traub conceived the MERCURY computer-aided distribution system [58]. Using subject codes from a hierarchical vocabulary together with organization and project numbers and individual names, authors describe the readers they wish to reach, and readers describe the papers they wish to receive. These descriptions are matched by a computer, which prints a distribution list and addressing labels for each paper. Developed in cooperation with the Bell Labs library, MERCURY went into service in April 1966. Besides MERCURY, the library has developed many other computer-aided systems to provide information services and support for the library network, including the BELLREL system for real-time management of the book and journal collections, the BELLTIP system for book ordering and cataloging, the BELLPAR and BELLTAB system for producing current awareness bulletins, and the BELDEX system for constructing specialized indexes, catalogs, and bibliographies [59].

Sometimes a computer can be used to store large quantities of data for subsequent analysis. An early example was the reduction, storage, retrieval, analysis, and display (discussed above) of data on the earth's radiation belts, measured by the TELSTAR satellites.

Frequently, the entire data collection must be instantly accessible at many widely separated locations to users who may wish to store in it or retrieve from it or both. To support such applications, Norman R. Sinowitz at Bell Labs developed an interactive information retrieval system called DATAPLUS [60]. This work was augmented through the provision of a general-purpose data-management system called Master Links and a generalized interactive-dialogue system called the Natural Dialogue System [61]. These, in turn, were combined and augmented to furnish a packaged information-management system, called the Off-The-Shelf system [62]. Finally, for telephone directory information, Michael E. Lesk developed an experimental Bell Labs directory-assistance system on a minicomputer, enabling a caller to type the last name and initials of a fellow employee on a TOUCH-TONE telephone, and receive the called party's extension by voice response [63]. In less than 5 percent of all cases, the request is ambiguous, and the caller is given a list of alternatives.

12. COMPUTERS AND MATHEMATICS

Although we have so far considered many aspects of computer science with hardly a mention of mathematics, the relationship between the two disciplines is intimate and multifaceted. Like mathematics, computer science is not only a rich and fascinating subject in its own right, but also provides language and tools for all other sciences. While the role of computing within mathematics has always been fundamental, the role of mathematics in computer science is perhaps equally pervasive.

Looking first at computers themselves, we find that their logical design is described in terms of Boolean algebra in accordance with principles discovered by Shannon in 1937 (see Section 3), shortly before he began his distinguished career at Bell Labs. About a decade later, Shannon formulated the mathematical theory of information, and established the use of the binary digit, or bit as the standard measure of information. The name bit was suggested by John W. Tukey of Bell Labs, and is routinely used in specifying the size of computer memories, data transmission rates, etc. The reliable operation of computers depends on error-detecting codes, first invented around 1938 by Ralph E. Hersey for use in telephone switching offices, and introduced to computing in 1942 by Stibitz to enhance the reliability of the Model II relay machine. Later, in 1948, Hamming extended this idea to the development of error-correcting codes, and thereby founded the branch of mathematics that is now known as algebraic coding theory.

In studying the ultimate capabilities and limitations of computers, Mealy and Edward F. Moore, both of Bell Labs, introduced abstract models that provided significant impetus to the emerging mathematical theory of automata .66 Concurrently, the rise of programming languages led to the development of mathematical linguistics, and it was later shown that the two fields are essentially one and the same.

Subsequent investigations led Alfred V. Aho and Jeffrey D. Ullman of Bell Labs to important theoretical advances in the then rapidly developing field of formal language theory. Later, joined by Stephen C. Johnson, also of Bell Labs, these investigators extended the applicability of a powerful parsing technique from formal language theory to ambiguous grammars. 68 The broad utility of this approach, together with its good error-detecting properties, enabled Johnson to successfully employ this technique in a program generator called YACC (Yet Another Compiler Compiler), which has proven useful in a surprisingly wide variety of applications [69].

Paralleling these advances in formal languages was the development of algorithms for translating parser output into optimal sequences of machine instructions. In 1969, Ullman and Ravi Sethi developed such an algorithm at Bell Labs for arithmetic expressions on machines with simple instruction sets [70]. At about the same time, Stephen G. Wasilew used dynamic programming in a code generator for the ESS programming language (EPL) for ESS machines [71]. Later, Aho and Johnson merged and extended these separate approaches to obtain a general algorithm for a broad class of machines. To produce an optimal program, or even a good one, it is not sufficient to deal correctly with each constituent expression and statement; global considerations are also crucial. In 1961, Vyssotsky conceived an efficient algorithm for global data flow analysis, and used it to provide an advanced diagnostic capability in the Bell Labs compiler for FORTRAN II on the IBM 7090.

Of course, computers were originally built for the purpose of solving mathematical problems. Their spectacular successes have stimulated a great effort to develop efficient algorithms for recurring mathematical tasks and to make them readily available as library procedures or as basic operations in mathematically oriented languages. Although numerical analysis was probably the first branch of mathematics to be studied from this point of view, the goal of getting results of provably high quality at a reasonable cost is still a topic for current research. Among the important contributions of Bell Labs mathematicians to numerical analysis in the early 1960s were a thorough study by Traub of the complexity of a large family of iterative numerical algorithms, and the clear recognition by Hamming that the interests, tastes, and objectives of practicing numerical analysts are necessarily quite different from those of most other mathematicians.

Similarly, many statisticians in the early 1960s found themselves motivated more by the desire to understand their data than by the criteria of other mathematicians, and Tukey coined the phrase data analysis to characterize their emerging discipline [75]. Inspired by Tukey, Bell Labs statisticians, including Martin B. Wilk, John D. Gabbe, and John M. Chambers, pioneered in the use of computers for storing, retrieving, and analyzing very large sets of data. Rapidly improving computer-output capabilities (see Section 11) spurred the development of probability plotting methods in the middle 1960s, and the introduction of interactive color displays (see Fig. 15) in the early 1970s for contour-type plotting and a variety of other scientific, technical, and artistic applications.

One very common type of statistical computation is the Monte Carlo simulation of a process in which statistics are collected on a large number of trials controlled by random numbers. This technique was developed about 1920 by Molina (the inventor of the relay translator, as noted in Section 3), so that he could simulate telephone traffic in a proposed network and thereby optimize the design of Bell System central offices. These simulations were called *throwdowns* because dice were literally thrown down to get the random numbers. The necessary computing and collection of statistics were carried out by clerks, whose instructions would now be viewed as a computer program.

Thanks to the development of electronic computers with large high-speed memories, Monte Carlo simulations soon became a very important research and design technique. Among the notable advances at Bell Labs in the early 1960s were the Sequence Diagram Simulator (SDS) designed by John P. Runyon, Donald L. Dietmeyer, Geoffrey Gordon, and Berkley A. Tague and the NEtwork Analytical SIMulator (NEASIM) designed by Richard F. Grantges and Sinowitz [78].

Both in numerical analysis and in data analysis, one of the most common tasks is the computation of the discrete Fourier transform. This was often prohibitively time-consuming until 1965 when James W. Cooley of IBM and Tukey of Bell Lab [79]. (and, independently, Gordon Sande of Princeton [80]) developed and made known an algorithm for the purpose, commonly called the Fast Fourier Transform (FFT), which was later found to have a variety of precursors [81].

Various forms of the FFT algorithm spawned the development of a family of special-purpose digital FFT processors. The cascade (or pipeline) architecture was developed in 1966 by G. David Bergland and Richard Klahn [82]. Then, in 1967, Richard R. Shively and his associates completed the first sequential FFT processor (see Figs. 16 and 17), which was used for research in digital signal processing [83]. Finally, in 1969, Bergland and Donald E. Wilson introduced a new version of the algorithm, suitable for implementation on computers employing multiple processors in parallel [84].

Because of their universality, computers are perfectly capable of deriving symbolic mathematical expressions as well as numbers. Since symbolic results are free of round-off error and may provide more insight as well, Brown, Tague, and John P. Hyde of Bell Labs developed the ALPAK package of subroutines for symbolic algebra in the early 1960s [85]. Then, in the middle 1960s, Brown, McIlroy, Gerald S. Stoller, and Leagus developed the ALTRAN language to facilitate ALPAK programming. 86 Shortly after the completion of the ALTRAN translator, the IBM 7094 computers, on which ALPAK and ALTRAN were totally dependent, began to be replaced by newer machines. This seemingly unfortunate situation led to a more advanced ALTRAN language and system developed by Brown, Hall, Johnson, Dennis M. Ritchie, and Stuart I. Feldman, which is highly portable (see Section 9) and has proven useful in a wide variety of scientific applications, both at Bell Labs and elsewhere [87]. Later, Feldman and Julia Ho added a rational expression evaluation package that generates accurate and efficient FORTRAN subroutines for the numerical evaluation of symbolic expressions produced by ALTRAN [88].

One of the central problems in symbolic algebraic systems such as ALTRAN is the computation of the greatest common divisors of polynomials. Early attempts to generalize Euclid's algorithm for this purpose encountered serious computational obstacles, which were overcome in the early 1970s with the aid of basic contributions by Brown and Traub [89]. Similar obstacles to polynomial factoring were overcome at about the same time with a fundamental algorithm devised by Elwyn R. Berlekamp [90].

Bell Labs mathematicians have contributed basic computer algorithms for other areas of nonnumerical mathematics as well. In 1956, Kruskal presented a simple, elegant algorithm for finding a minimal spanning tree in a graph with edges of specified lengths [91]. For *dense* graphs, with a high ratio of edges to nodes, Robert C. Prim provided a more efficient procedure in 1957 [92]. An efficient algorithm to generate all the spanning trees in a graph was given by McIlroy in 1969 [93].

An important problem in graph theory, called the traveling salesman problem, is to find the shortest closed path through all the nodes of a graph. For large graphs, no efficient algorithm is known, and it is believed that none exists. However, in the late 1960s, Shen Lin and Brian W. Kernighan of Bell Labs invented a number of increasingly powerful heuristic methods, which produce generally good solutions that are often optimal [94].

In studying the problem of assigning tasks to multiple processors, Ronald L. Graham of Bell Labs was perhaps the first to analyze the quality of solutions generated by such techniques [95]. More recently, Graham and his associates have provided similar analyses for a number of other computationally difficult problems [96]. Perhaps the most ambitious goal in the application of computers to mathematics is automated theorem proving. An early milestone, achieved by Hao Wang of Oxford University during his sabbatical visit to Bell Labs in the academic year 1959-1960, was the development of a program that proved all of the more than 350 theorems of first-order predicate calculus from the *Principia Mathematica* in only 8.4

minutes on an IBM 704 computer [97].

While the attempt to mechanize mathematics has raised many fascinating new. mathematical questions, its successes have created new mathematical opportunities. As *objects* of mathematical study, the computer and its languages have opened up new realms of fruitful investigation. As *tools* for mathematical study, they have permitted mathematics to evolve into an experimental science and at the same time have helped mathematicians to prove theorems more easily. In both roles, computers have shifted the emphasis in mathematics from static theorems to dynamic algorithms and have thereby contributed to a deeper appreciation of the rich structures that were always there. Finally, they have fundamentally altered the real world to which mathematics must ultimately relate, and have provided new ways in which that relationship may occur.

References

1. C. A. Lovell and L. E. Kittredge, "First Crossbar Throwdown-Terminating Train," unpublished, November 1937.

2. B. D. Holbrook and J. T. Dixon, "Load Rating Theory for Multi-Channel Amplifiers," Bell System Technical J.18 (October 1939), pp. 624-644.

3. S. B. Wright and E. R. Taylor, "Volume Control in Telephone Circuits," U. S. Patent No. 1,927,999, filed September 1929, issued May 1933.

4. The Development and Research Department of AT&T became part of Bell Labs in 1934.

5a. H. S. Black, "Stabilized Feedback Amplifiers," Bell System Technical J. 13 (January 1934), pp. 1-18.

5b. H. W. Bode, Network Analysis and Feedback Amplifier Design, D. Van Nostrand, New York, 1945.

6. E. C. Molina, U.S. Patent No. 1,083,456, filed April 1906, issued January 1914.

7. C. E. Shannon, "A Symbolic Analysis of Relay and Switching Circuits," Transactions of the AIEE 57 (December 1938), pp. 713-723.

8. For a general account of the planimeter, see the Encyclopaedia Brittanica, 1970 ed., vol. 14, p. 1085.

9. William Thomson (Lord Kelvin), Mathematical and Physical Papers, vol. 6 (Cambridge University Press, 1911), p. 287, as quoted from the Catalogue of the Special Loan Collection of Scientific Apparatus at the South Kensington Museum, 1876, p. 11.

10a. R. L. Dietzold, "The Isograph-A Mechanical Root-Finder," Bell Laboratories Record 16 (December 1937), pp. 130-134.

10b. R. O. Mercner, "The Mechanism of the Isograph," Bell Laboratories Record 16 (December 1937), pp. 135-140.

11. C. A. Lovell, "Continuous Electrical Computation," Bell Laboratories Record 25 (March 1947), pp. 114-18.

12. A. A. Currie, "The General Purpose Analog Computer," Bell Laboratories Record 29 (March 1951), pp. 101-108.

13. E. G. Andrews, "Bell Laboratories Digital Computers," Bell Laboratories Record 35 (March 1957), pp. 81-84.

14. S. B. Williams, "Remotely Controlled Electrical Calculator," U.S. Patent No. 2,434,681, filed February 1943, issued January 1948.

15. E. G. Andrews, "Telephone Switching and the Early Bell Laboratories Computers," Bell System Technical J. 42 (March 1963), pp. 341-353.

16. Franz L. Alt, "A Bell Telephone Laboratories Computing Machine," Mathematical Tables and Aids to Computation 3 (January and April 1948), pp. 1-13 and 69-84.

17. E. G. Andrews, "The Bell Computer Model VI," Electrical Engineering 68 (September 1949), pp. 751-756.

18. T. L. Dimond, "No. 5 Crossbar AMA Translator," Bell Laboratories Record 29 (February 1951), pp.62-68.

19. It is anticipated that more detailed information on automatic message accounting will be included in a subsequent volume of this history.

20. G. R. Frost, W. Keister, and A. E. Ritchie, "A Throwdown Machine for Telephone Traffic Studies," Bell System Technical J. 32 (March 1953), pp.292-359.

21. Brian Randell (Ed.), The Origins of Digital Computers, Springer-Verlag, New York, 1975. An excellent summary of the work of Zuse and others begins on p. 155.

22. H. H. Aiken et al., A Manual of Operation for the Automatic Sequence Controlled Calculator, Annals of the Computation Laboratory of Harvard University, Vol. 1, Harvard University Press, Cambridge, Mass., 1946.

23. "Description of a Relay Calculator," Annals of the Computation Laboratory of Harvard University, Vol. 24, Harvard University Press, Cambridge, Mass., 1949.

24. H. H. Goldstine and A. Goldstine, "The Electronic Numerical Integrator and Computer (ENIAC)," Mathematical Tables and Aids to Computation 2 (July 1946), pp. 97-110.

25. W. H. T. Holden, U.S. Patent No. 2,299,898, filed October 1941, issued October 1942.

26. A. W. Horton, U.S. Patent No. 2,244,700, filed September 1939, issued June 1941.

27. W. H. MacWilliams, Jr., "A Transistor Gating Matrix for a Simulated Warfare Computer," Bell Laboratories Record 35 (March 1957), pp. 94-99.

28. V. M. Wolontis, "A Complete Floating-Decimal Interpretive System for the IBM 650 Magnetic Drum Calculator," IBM Technical Newsletter, No. 11, March 1956.

29. F. J. Corbato, M. M. Daggett, and R. C. Daley, "An Experimental Time Sharing System," Proceedings of the Spring Joint Computer Conference of the American Federation of Information Processing Societies, 1962, pp. 335-344.

30. F. J. Corbato, V. A. Vyssotsky, et al., "A New Remote Accessed Man-Machine System," Proceedings of the Fall Joint Computer Conference of the American Federation of Information Processing Societies, 1965, Part I, pp. 185-247.

31. S. P. Morgan, "Minicomputers in Bell Laboratories Research," Bell Laboratories Record 51 (July/August 1973), pp. 194-201.

32a. K. Thompson and D. M. Ritchie. "UNIX Time Sharing System," Communications of the ACM 17 (July 1974), pp. 365-375.

32b. Special UNIX Issue, Bell System Technical Journal, July-August 1978, Part 2.

33. M. D. McIlroy, "Macro Instruction Extension of Compiler Languages," Communications of the ACM 3 (April 1960), pp. 214-220.

34. N. A. Martellotto, H. Oehring, and M. C. Paull, "PROCESS III: A Compiler-Assembler for No. 1 ESS," Bell System Technical J. 43 (September 1964), pp. 2457-2481.

35. M. E. Barton, "The Macro Assembler SWAP--A General-purpose Interpretive Processor," Proceedings of the Fall Joint Computer Conference of the American Federation of Information Processing Societies, 1970, pp. 1-8..

36. B. N. Dickman, "CENTRAN: A Case History in Extendible Language Design," Bell System Technical J. 54, SAFEGUARD Special Supplement (May 1975), pp. S161-S172.

37. J. L. Kelly, Jr., C. C. Lochbaum, and V. A. Vyssotsky, "A Block Diagram Compiler," Bell System Technical J. 40 (May 1961), pp. 669-676.

38. K. C. Knowlton, "A Programmer's Description of L6," Communications of the ACM 9 (August 1966), pp. 616-625.

39. E. N. Pinson and C. Christensen, "Multi-Function Graphics for a Large Computer System," Proceedings of the Fall Joint Computer Conference of the American Federation of Information Processing Societies, 1967, pp. 697-711.

40. M. V. Mathews, The Technology of Computer Music, M.I.T. Press, Cambridge, Mass., 1969.

41. D. J. Farber, R. E. Griswold, and I. P. Polonsky, "SNOBOL: A String Manipulation Language," J. of

the ACM 11 (January 1964), pp. 21-30.

42. R. E. Griswold, J. F. Poage, and I. P. Polonsky, The SNOBOL4 Programming Language, 2nd ed., Prentice-Hall, Englewood Cliffs, N.J., 1971.

43. R. E. Griswold, The Macro Implementation of SNOBOL4: A Case Study of Machine-Independent Software Development, W. H. Freeman, San Francisco, 1972.

44. B. G. Ryder, "The PFORT Verifier," Software Practice and Experience 4 (October/December 1974), pp. 359-377.

45. D. M. Ritchie and B. W. Kernighan, The C Programming Language, Prentice Hall, 1978.

46. Trademark of The Singer Company.

47. W. D. Farmer and E. E. Newhall, "An Experimental Distributed Switching System to Handle Bursty Computer Traffic," Proceedings of the ACM Symposium on Problems in the Optimization of Data Communications Systems, Pine Mountain, Georgia, October 1969, pp. 1-34.

48. J. R. Pierce, "Network for Block Switching of Data," Bell System Technical J. 51 (July/August 1972), pp. 1133-1145.

49. A. G. Fraser, "Spider-A Data Communications Experiment," Computing Science Technical Report No. 23, Bell Laboratories, Murray Hill, N.J., December 1974.

50. A. R. Breithaupt, "Project Viperidae: A Bell Labs Computing Network," COMPCON 73 Digest of Papers, IEEE Computer Society International Conference, February 1973, pp. 235-238.

51. J. F. Kaiser, "Graphs Should be Computer Drawn," in The Human Use of Computing Machines, Bell Telephone Laboratories, Murray Hill, N.J., June 1966, pp. 9-14.

52. W. L. Brown and J. D. Gabbe, "The Electron Distribution in the Earth's Radiation Belts During July 1962 as Measured by Telstar®, J. of Geophysical Research 68 (February 1963), pp. 607-618.

53. E. E. Zajac, "Computer-Made Perspective Movies as a Scientific and Communication Tool," Communications of the ACM 7 (March 1964), pp. 169-170.

54a. A. M. Noll, "Computer-Generated Three-Dimensional Movies," Computers and Automation 14 (November 1965), pp. 20-23.

54b. F. W. Sinden, "Synthetic Cinematography," Perspective 7, 1965, pp. 279-289.

55. K. C. Knowlton, "Computer-Produced Movies," Science 150 (November 1965), pp. 1116-1120.

56. W. H. Ninke, "GRAPHIC I: A Remote Graphical Display Console System," Proceedings of the Fall Joint Computer Conference of the American Federation of Information Processing Societies 27 (1965), Part I, pp. 839-846.

57. See note 39, above.

58. W. S. Brown and J. F. Traub, "MERCURY: A System for the Computer-Aided Distribution of Technical Reports," Journal of the ACM 16 (January 1969), pp. 13-25.

59a. W. K. Lowry, "Use of Computers in Information Systems, " Science 175 (February 25, 1972), pp. 841-846.

59b. R. A. Kennedy, "Bell Laboratories Library Real-Time Loan System (BELLREL)," Journal of Library Automation 1 (June 1968), pp. 128-146.

60. N. R. Sinowitz, "DATAPLUS: A Language for Real Time Information Retrieval from Hierarchical Data Bases," Proceedings of the Spring Joint Computer Conference of the American Federation of Information Processing Societies 32 (1968), pp. 395-401.

61a. T. A. Gibson and P. F. Stockhausen, "MASTER LINKS: A Hierarchical Data System," Bell System Technical J. 52 (December 1973), pp. 1691-1724.

61b. B. W. Puerling and J. T. Roberto, "The Natural Dialogue System," Bell System Technical J. 52 (December 1973), pp. 1725-1741.

62. L. E. Heindel and J. T. Roberto, "The Off-The-Shelf System: A Packaged Information Management System," Bell System Technical J. 52 (December 1973), pp. 1743-1763.

63. Reference 31, page 200.

64. C. E. Shannon, "A Mathematical Theory of Communication. Bell System Technical J. 27 (July 1948), pp. 379-423 and (October 1948), pp. 623-656. (See esp. p.380.)

65. R. W. Hamming, "Error Detecting and Error Correcting Codes, Beg System Technical J. 29 (April 1950), pp. 147-160.

66a. G. H. Mealy, "A Method for Synthesizing Sequential Circuits," Beg System Technical J. 34 (September 1955), pp. 1045-1079.

66b. E. F. Moore, "Gedanken Experiments on Sequential Machines," in Automata Studies (eds. C. E. Shannon and J. McCarthy), Princeton University Press, Princeton, N.J., 1956, pp. 129-153.

67. A. V. Aho and J. D. Ullman, The Theory of Parsing, Translation and Compiling, 2 vols., Prentice-Hall, Englewood Cliffs, NJ., 1972 and 1973.

68. A. V. Aho, S. C. Johnson, and J. D. Ullman, "Deterministic Parsing of Ambiguous Grammars," Communications of the ACM 18 (August 1975), pp. 441-458.

69. S. C. Johnson, "YACC: Yet Another Compiler Compiler," Computing Science Technical Report No. 32, Bell Laboratories, Murray Hill, N.J., July 1975.

70. R. Sethi and J. D. Ullman, "The Generation of Optimal Code for Arithmetic Expressions," Journal of the ACM 17 (October 1970), pp. 715-728.

71. S. G. Wasilew, "A Compiler Writing System with Optimization Capabilities for Complex Order Structures," Ph.D. diss., Northwestern University, 1971.

72. A. V. Aho and S. C. Johnson, "Optimal Code Generation for Expression Trees," Proceedings of the 7th Annual ACM Symposium on Theory of Computing, May 1975, pp. 207-217.

73. M. S. Hecht and J. D. Ullman, "Analysis of a Simple Algorithm for Global Data Flow Problems," Proceedings of the ACM Symposium on Principles of Programming Languages. October 1973, pp. 207-217. (See esp. p.207.)

74a. J. F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, N.J., 1964.

74b. R. W. Hamming, "Numerical Analysis vs. Mathematics," Science 148 (April 1965), pp. 473-475.

75. J. W. Tukey, "The Future of Data Analysis," Annals of Mathematical Statistics 33 (March 1962), pp. 1-67.

76a. J. D. Gabbe, M. B. Wilk, and W. L. Brown, "Statistical Analysis and Modeling of the High-Energy Proton Data from the Telstar® I Satellite," Beg System Technical J. 46 (September 1967), pp. 1301-1450.

76b. J. M. Chambers, "A Statistical Data Language," in Statistical Computation (eds. R. Milton and J. A. Nelder), Academic Press, New York, 1969, pp. 179-199.

77a. M. B. Wilk and R. Gnanadesikan, "Probability Plotting Methods for the Analysis of Data," Biometrika 55 (March 1968), pp. 1-17.

77b. P. B. Denes, "Computer Graphics in Color," Beg Laboratories Record 52 (May 1974), pp. 139-146.

78a. D. L. Dietmeyer, G. Gordon, J. P. Runyon, and B. A. Tague, "An Interpretive Simulation Program for Estimating Occupancy and Delay in Traffic-Handling Systems Which Are Incompletely Detailed," AIEE Conference Paper 60-1090, Pacific General Meeting, August 1960 (unpublished).

78b. R. F. Grantges and N. R. Sinowitz, "NEASIM: A General Purpose Computer Simulation Program for Load-Loss Analysis of Multistage Central Office Switching Networks," Beg System Technical J. 43 (May 1964), pp. 965-1004.

79. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation 19 (April 1965), pp.297-301.

80. C. Bingham, M. D. Godfrey, and J. W. Tukey, "Modern Techniques of Power Spectrum Estimation," IEEE Transactions on Audio and Electroacoustics AU-15 (June 1967), pp. 5666.

81. J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "Historical Notes on the Fast Fourier Transform,"

Proceedings of the IEEE 55 (October 1967), pp.1675-1677.

82. G. D. Bergland and R. Klahn, "Digital Processor for Calculating Fourier Coefficients," U.S. Patent No. 3,544,775, filed December 1966, issued December 1970.

83. R. R. Shively, "A Digital Processor to Generate Spectra in Real Time," IEEE Transactions on Computers C-17 (May 1968), pp. 485-491.

84. G. D. Bergland and D. E. Wilson, "A Fast Fourier Transform Algorithm for a Global Highly Parallel Processor," IEEE Transactions on Audio and Electroacoustics AU-17 (June 1969), pp. 125-127.

85. W. S. Brown, J. P. Hyde, and B. A. Tague "The ALPAK System for Nonnumerical Algebra on a Digital Computer," Bell System Technical J.42, pp. 2081-2119, September 1963; 43, pp. 785-804, March 1964; 43, pp. 1547-1562, July 1964.

86. W. S. Brown, "A Language and System for Symbolic Algebra," System Analysis by Digital Computer (eds. F. F. Kuo and J. F. Kaiser), John Wiley, New York, 1966, pp. 349-369.

87. A. D. Hall, "The ALTRAN System for Rational Function Manipulation-A Survey," Communications of the ACM 14 (August 1971), pp. 517-521.

88. S. I. Feldman and J. Ho, "A Rational Expression Evaluation Package," Computing Science Technical Report No. 34, Bell Laboratories, Murray Hill, N.J., September 1975.

89a. W. S. Brown, "On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors," Journal of the ACM 18 (October 1971), pp. 478-504.

89b. W. S. Brown and J. F. Traub, "On Euclid's Algorithm and the Theory of Subresultants," J. of the ACM 18 (October 1971), pp. 505-514.

90. E. R. Berlekamp, "Factoring Polynomials over Large Finite Fields," Mathematics of Computation 24 (July 1970), pp. 713-735.

91. J. B. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," Proceedings of the American Mathematical Society 7 (February 1956), pp. 48-50.

92. R. C. Prim, "Shortest Connection Networks and Some Generalizations," Bell System Technical J36 (November 1957), pp. 1389-1401.

93. M. D. McIlroy "Algorithm 354: Generator of Spanning Trees," Communications of the ACM 12 (September 1969), p. 511.

94. S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," Operations Research 21 (March/April 1973), pp. 498-516.

95. R. L. Graham, "Bounds for Certain Multiprocessing Anomalies," Bell System Technical J. 45 (November 1966), pp. 1563-1581.

96. D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms," SIAM Journal on Computing 3 (December 1974), pp. 299-325.

97a. Hao Wang, "Proving Theorems by Pattern Recognition, I," Communications of the ACM 3 (April 1960), pp. 220-234.

97b. A. N. Whitehead and B. Russell, Principia Mathematica, 2nd ed., Cambridge University Press, 1927.

Index of Authors and Researchers

Name	**Section(s)**	**Ref(s)**
Graham, Ronald L.	12	95,96
Grampp, Frederick T.	8	
Grantges, Richard F.	12	78
Griswold, Ralph E.	9	41-43
Hall, Andrew D.	9,12	44,87
Hamming, Richard W.	4,7,12	65,74
Hecht, Matthew S.	12	73
Heindel, Lee E.	11	62
Hersey, Ralph E.	12	
Ho, Julia	12	88
Holbrook, Bernard D.	2	2
Holden, William H. T.	6	25
Horton, Arthur W., Jr.	6	26
Hyde, John P.	12	85
Johnson, David S.	12	96
Johnson, Stephen C.	12	68,69,72,87
Kaiser, James F.	11	51
Keister, William	5.5	20
Kelly, John L. ,Jr.	9	37
Kelvin, Lord	4	9
Kennedy, Robert A.	11	59
Kernighan, Brian W.	9,12	45,94
Kittredge, Linus E.	2	1
Klahn, Richard	12	82
Knowlton, Kenneth C.	9,11	38,55
Kruskal, Joseph B.	11,12	91
Lakatos, Emory	4	12
Leagus, Dolores C.	7,12	28,86
Lesk, Michael E.	9,11	45,63
Lewis, Peter A. W.	12	81
Lin, Shen	12	94
Lochbaum, Carol C.	9	37
Lovell, Clarence A.	2,4	1,11
Lowry, W. Kenneth	11	59
MacWilliams, Walter H.	6	27
Martellotto, Nicholas A.	9,10	34,50
Mathews, Max V.	9	40
Mauchly, John W.	5.6	24
McCarthy, John	12	66
McClure, Robert M.	11	
McDonald, Henry S.	8,11	56
McIlroy, M. Douglas	9,12	33,93
Mealy, George H.	12	66
Mercner, Raymond O.	4	10
Michelson, Albert A.	4	10
Molina, Edward C.	3,12	6
Moore, Edward F.	12	66
Morgan, Samuel P.	8,11	31,63
Newhall, Edwin E.	10	47
Ninke, William H.	11	56,57
Noll, A. Michael	11	54
Oehring, Hans	9	34

Index of Events

Date	Section (s)	Ref(s)	Event
Before 1920			
1854	4	8	Polar planimeter invented.
1876	4	9	Ball-and-disk integrator used to predict tides.
1906	3	6	Relay translator invented.
The 1920s			
Early 1920s			
	12		Throwdowns to simulate telephone traffic.
Late 1920s			
1927	3	5	Feedback amplifier invented.
1928	1		Desk calculators and slide rules used for design calculations.
	2		Punched-card equipment used for cost accounting.
1929	3	3	Analog adder invented for radiotelephone control.
The 1930s			
Early 1930s			
	4		Electrical control added to mechanical analog computers.
	4		Mechanical analog computers used for gun control.
Middle 1930s			
	2	1	Punched-card equipment used for traffic-congestion problems.
	6	25	AND gate invented.
	6	26	OR gate invented.
1936	5.6	21	Electrical digital computer proposed by Zuse.
Late 1930s			
1937	3,12	7	Boolean algebra used for circuit design.
	4	10	Isograph mechanical analog computer developed.
	5.1		Complex Number Computer conceived.
1938	2	2	Analog adders used to design multi-channel amplifiers.
	3	3	Analog adder used to control radio-telephone facilities.
	12		Error-detecting codes invented.
1939	5.1	13	Complex Number Computer completed; later called the Model 1.
The 1940s			
Early 1940s			
1940	4	11	Electrical analog computer invented.
	5.1	14	Complex Number Computer demonstrated remotely.
1942	4		M-9 gun director delivered to the Army.
	5.2	15	Paper-tape input introduced in the Model II relay computer.
	12		Error-detecting codes introduced in the Model II relay computer.
1943	5.2	15	Model II relay computer placed in service.
Middle 1940s			
	5.6	23	Mark II electro-mechanical computer developed by Aiken.
1944	4		M-9 gun director used in Britain.
	5.2	15	Model III relay computer placed in service.
1945	5.2	16	Model IV relay computer placed in service.
	5.6	22	Mark I electro-mechanical computer completed by Aiken.
1946	5.3	16	Model V relay computer delivered to the NACA.
	5.6	24	ENIAC vacuum-tube computer completed by Mauchly and Eckert.
	5.6		Stored-program computer conceived by Von Neumann.

Date	Section(s)	Ref(s)	Event
Late 1940s			
1947	5.3		Second Model V relay computer delivered to the BRL.
	6		Transistor effect discovered.
1948	5.5	19	Relay computer developed for automatic message accounting.
	12	64	Shannon's mathematical theory of information.
	12	64	Tukey coins the word "bit".
	12	65	Error-correcting codes invented; algebraic coding theory founded.
1949	4	12	General Purpose Analog Computer developed.
	5.1		Complex Number Computer retired.
	6	27	Transistor gating matrix for simulated-warfare computer.
The 1950s			
	5.5		Relay computers used for automatic message accounting.
	6		Transistor computers developed for the armed forces.
Early 1950s			
	5.5	20	Relay computer developed for traffic-congestion problems.
	6		Time rented on commercial computers.
1950	5.4	17	Model VI relay computer placed in service.
	5.4	18	Dimond-ring translators used for semipermanent subroutines.
	5.6		EDVAC vacuum-tube computer placed in service.
1952	6		IBM Card Programmed Calculator acquired.
Middle 1950s			
	10		Tape-to-tape transmission of digital data.
1955	6		IBM 650 acquired.
	7	28	L1 language introduced (also known as Bell 1).
	12	66	Mealy's abstract model of computers.
1956	5.4		Model VI relay computer to Brooklyn Polytechnic Institute
	6		Second IBM 650 acquired.
	7		L2 language introduced (also known as Bell 2).
	10		Data transmission by telephone.
	2	66	Moore's abstract model of computers.
	12	91	Kruskal's algorithm for minimal spanning trees.
Late 1950s			
	6		Magnetic cores and crystal diodes used in vacuum-tube computers.
	7		Bell 1 and Bell 2 widely used on IBM 650s.
1957	7		IBM 704 acquired.
	8		BESYS-2 placed in service.
	9		Symbolic assembly languages with macros.
	12	92	Prim's algorithm for minimal spanning trees.
1959	8		IBM 704 installed at Whippany.
	9	33	Conditional and recursive macros introduced.

Date	Section(s)	Ref(s)	Event
The 1960s			
Early 1960s			
	9	41	SNOBOL language.
	10		Remote computing service for branch laboratories.
	11		Microfilm package.
	11	52	Plots of earth’s radiation belts.
	11		Movie of incoming missiles and decoys.
	11		Movie of multidimensional scaling iterations.
	12	74	Complexity studies of iterative numerical algorithms.
	12	74	Numerical analysis contrasted with classical mathematics.
	12	75	Tukey coins the phrase "data analysis."
	12	85	ALPAK system for symbolic algebra.
1960	4		GPAC analog computers to Brooklyn Polytechnic Institute
	4		Commercial analog computer acquired.
	5.4		Model VI relay computer arrives at the University of Bihar.
	8		BESYS-3 placed in service.
	9	33	Macros used for language extension.
	12	78	Sequence Diagram Simulator.
	12	97	Theorem proving programs for first-order predicate calculus.
1961	9	37	BLODI language.
		11	
	12	73	Vyssotsky’s algorithm for global data flow analysis.
1962	8		BESYS-4 placed in service.
	8	29	Compatible Time Sharing Service developed at MIT.
	10		Computing center at Holmdel served from Murray Hill.
	10		Data transmission via TELSTAR" satellite demonstrated.
1963	8		BESYS-5 placed in service.
	10		1.5 megabaud link between Murray Hill and Holmdel installed.
Middle 1960s			
	12	77	Probability plotting methods developed.
	12	86	Early ALTRAN language and system.
1964	8		BESYS-7 placed in service.
	8	30	Research on MULTICS begun.
	8		Work on TSS begun.
	9	34	Process III assembler.
	11	53	Movie of satellite motion.
	11		PB 250 used as intelligent terminal.
	12	78	Network Analytic Simulator
1965	11	54	Educational movie.
	11	54	Stereographic three-dimensional movie.
	11	55	BEFLIX language.
	11	56	GRAPHIC-1 terminal.
	12	79-81	FFT algorithm developed.
1966	9	38	L6 language.
	10		Computing center at Murray Hill served from Holmdel.
	11	51	TPLOT subroutine for drawing graphs.
	11	58	MERCURY system for report distribution placed in service.
	12	82	Cascade architecture for FFT processors.
	12	95	Analyses of heuristic solutions to assignment problems.

Date	Section(s)	Ref(s)	Event
Late 1960s			
	8	31	Pioneering uses of minicomputers.
	12	94	Heuristic methods for the traveling salesman problem.
1967	8		IBM 360/67 acquired at Indian Hill.
	8,10		GE 635/645 acquired at Murray Hill.
	9	39	GRIN language.
	11	57	GRAPHIC 2 terminal.
	12	76	Analysis of very large sets of data.
	12	83	Sequential FFT processor.
1968	8		BE90 emulator installed.
	10	47	Experimental loop network by Farmer and Newhall.
	11	59	BELLREL library circulation system placed in service.
	11	60	DATAPLUS system for interactive information retrieval.
1969	8		Last IBM 7094 at Bell Labs retired.
	8	32	UNIX operating system developed.
	9	40	MUSIC-V language.
	12	70	Algorithm for optimal code generation on simple machines.
	12	71	Code generator for the EPL language for ESS computers.
	12	76	Statistical data language.
	12	84	Parallel FFT algorithm.
	12	93	Algorithm to generate all the spanning trees of a graph.
The 1970s			
Early 1970s			
	9	45	C language.
	12	77	Advances in formal language theory.
	12	77	Interactive color displays.
	9,12	87	Revised ALTRAN language and portable implementation.
1970	8		IBM TSS fully operational at Indian Hill.
	9	35	SWAP assembler.
	10	48	Loop network proposed by Pierce.
	10	49	SPIDER network proposed by Fraser.
	12	89	Polynomial GCD algorithm.
	12	90	Polynomial factoring algorithm.
1971	9	36	CENTRAN compiler (also known as ETC).
	9	42	SNOBOL-4 language.
1972	8		BE90 emulator retired.
	9	43	Portable SNOBOL-4 processor.
	10	50	Bell Labs Interlocation Computing Network proposed.
1973	11	61	Master Links data management system.
	11	61	Natural Dialogue system for interactive dialogue.
	11	62	Off-the-Shelf system for developing IMSs.
	11	31	Experimental automated directory assistance system.
Middle 1970s			
	11		TROFF language.
1974	9	44	PFORT Verifier.
	10		Bell Labs Interlocation Computing Network completed.
	12	96	Analyses of heuristic solutions to packing problems.
1975	12	68	Formal language theory applied to ambiguous grammars.
	12	69	YACC parser generator developed.
	12	72	General algorithm for optimal code generation.
	12	88	Rational expression evaluation package for ALTRAN.