Howard Aiken and the Beginnings of Computer science

I. Bernard Cohen
emeritus Victor S. Thomas - professor in history of science, Harvard University

Introduction

Howard Aiken's high place in computer history is given in the standard *Encyclopedia of Computer Science and Engineering*:

"The digital computer age began when the Automatic Sequence Controlled Calculator [Harvard Mark I] started working in April 1944." Another article in the same *Encyclopedia* begins: "The Harvard Mark I, also called the IBM Automatic Sequence Controlled Calculator... marked the beginning of the era of the modern computer." In 1964, AFIPS (the American Federation of Information Processing Societies) established the Harry Goode Memorial Award to honor its second president, Harry H. Goode, by recognizing "outstanding achievement in the field of information processing": the first award (1964) went to the recognized pioneer, the inventor whose giant machine had inaugurated the computer age: Howard H. Aiken.

In order to appreciate the nature and magnitude of Aiken's achievement it is necessary to find out how his contributions were related to other developments of the same period and also how what he accomplished has been a force in getting to the present computer age. Aiken's career in computers began in the late 1930s and came into flower in the 1940s.

This first stage of computer history has some remarkable features that differentiate it from conventional history of science and technology. In both Germany and America in the 1940s, those who were developing the first computers or protocomputers were "new men" in the arena of science and technology, outsiders who had not been active or prominent members of any traditional "old boy" network. J.V. Atanasoff received some nominal support from Iowa State College, but his work was out of the mainstream of computer technology and was not very much known in wider circles. George Stibitz was a member of a large and well established research and development organization, the Bell Telephone Laboratories, but innovations in computing or machine calculation were at most a peripheral concern of his fellow scientists and engineers. Howard Aiken was a real outsider and upstart, only a graduate student in physics, for numerical calculation. J.P. Eckert and John Mauchly, working on ENIAC in the 1940s, were engineers in the Moore School, whose project seemed to be of no interest to those who were concerned with methods of practical computing. And in Germany, Konrad Zuse was a real "loner," outsider with no academic affiliation and no industrial support. His pioneering work in designing and building his first machines made use of scraps and second-hand spare parts. He constructed his first machine in the living room of his parents' home.

Early starters

In the 1920's, most scientists and engineers had no use for the many decimal places that would be made possible by an advanced automatic digital calculator; they were satisfied with the three-place accuracy provided by Vannevar Bush's Differential Analyzer and other analog devices. During this period the major use of tables was, as it had been in the 19th century, by astronomers and actuaries. In the 1930s, the great potential of business machines for scientific calculation was only just being realized. The real pioneer in this area was L.J. Comrie, a New Zealander established in London. Comrie, superintendent of the British Nautical Almanac office, recognized that commercially produced business machines had the capability of being linked together and used for scientific computing.

A fruitful line of development in the 1930s made use of punched-card machines. These had been developed in linear descent from the original data-processing punched-card machines invented by Herman Hollerith to handle the data accumulated in the 1890 census. Hollerith's company became part of IBM.

Here let me observe that punched-card machines are essentially cumulators. hence, since multiplication is a form of controlled addition, and since addition is successive cumulation, a punched-card statistical or business machine may easily be directed to perform additions and multiplications. In the IBM punched-card business machines, subtraction was performed as addition, using complement arithmetic.

The scientists associated with giant calculators or computing machines in the late 1930s were not directly concerned with problems arising from practical engineering needs or with military affairs. George Stibitz started out by amusing himself with the analogies of relay circuits and binary arithmetical operations; he was then led to design and build his first practical machine in order to facilitate computations with so-called complex numbers. In 1937, Stibitz was exploring the common features of binary arithmetic and electric circuits containing relays. He eventually designed a relay machine completed in October 1939 and put into regular use in January 1940; it was known as the "Complex Number Computer" (later shortened to "Complex Computer"). It could speedily and easily add, subtract, multiply, and divide complex numbers: "All it requires of the operator is that she type the problem correctly".

John V. Atanasoff, a professor at Iowa State College (now Iowa State University) was motivated by a concern with applied mathematics (systems of linear equations in engineering problems) to go beyond the limits of analog calculation and to explore the possibilities of digital calculating. A major problem for him was "the solution of systems of linear simultaneous algebraic equations". He was joined by Clifford Berry in 1939. The machine, intended to solve up to thirty simultaneous linear equations, was to be equipped with a *binary* card-punch and reader developed by Berry. A "prototype computing element" was demonstrated in autumn 1939, but the large-scale machine was never completed. The project was abandoned when Atanasoff left Iowa State to join the staff of the Naval Ordnance Laboratory in 1942, at the same time that Berry joined an engineering company in California.

The ABC achieved real fame during the famous 1971-73 court case, Honeywell v. Sperry Rand, when the presiding trial judge declared that "Eckert and Mauchly did not themselves first invent the automatic electronic digital computer but instead derived the subject matter from one Dr. John Vincent Atanasoff".

In Germany the beginnings of the modern computer centered on the work of Konrad Zuse. After graduation from the Technische Hochschule in Berlin Charlottenburg in 1935, Zuse started his career as a stress analyst for the Henschel Aircraft Company in Berlin. Even before graduation, he had begun to think about a universal calculator that would use binary arithmetic; so when his work at Henschel led to a dreary sequence of setting up and subsequently solving systems of simultaneous linear equations, Zuse —like Stibitz, Atanasoff, and Aiken— decided that the calculations could and should be mechanized. At the age of twenty-six, in 1936, he started to build his first machine.

Zuse's first two calculators were actually built in the living room of his parents' appartment with the help of friends. The first of these, the Z1, was a binary machine, controlled by punched tape; it had a mechanical memory consisting of a set of thin slotted metal plates, a thousand in number, in which the position of a pin in the slot would indicate a 0 or a 1. Although it was never fully operational, the Z1 convinced Zuse that he was on the right track. He then began collaboration with a gifted electrical engineer, Helmut Schreyer, with whom the produced the Z2. This machine was in many ways similar to the machines of Stibitz and Aiken in that it used telephone relays.

In 1942, Zuse began to construct the Z3, a machine that used vacuum tubes plus electro-mechanical relays. This machine may well be "the first operational general-purpose program-controlled calculator".

The Z3 was completely destroyed in an air attack on Berlin in April 1945. Zuse, undaunted began to build a larger version, the Z4. The Z3 and the Z4 did not influence the next stage of computer development, however, because information about them became known until too late to have any effect.

The most significant wartime development in computing was ENIAC (Electronic Numerical Integrator and Computer), designed and produced at the Moore School of the University of Pennsylvania under contract to the U.S. The two men responsible for the new idea and its implementation were John Mauchly and J. Prosper Eckert, the former a physicist, the latter an electronics engineer. The degree of their achievement can be gauged by the fact that ENIAC would be several orders of magnitude larger than any electronic device ever conceived or manufactured, and it would have to be reliable.

ENIAC marked a turning point in the history of computers by proving that supercalculators could be built with speedy vacuum tubes rather than slow relays and that the results would be reliable. By the time that ENIAC was taken out of service on 2 October 1955, it had probably done "more arithmetic than had been done by the whole human race prior to 1945".

Perhaps the most development during World War II was the Colossus series of machines produced for encryption and decoding in Britian. Many very talented British mathematicians, scientists, and engineers worked on this wartime project, among them Alan M. Turing, H.A. Newman, and I.J. Good. Only recently has full information about their activities come to light from behind the screen of the Official Secrets Act.

Despite their priority with respect to electronics and programming, the Colossus machines did not have any open and direct influence on the later developments of computers because they were shrouded in secrecy until the 1980s. Yet some of the scientists and engineers associated with Bletchy Park — e.g. Newman, Flowers, Turing— later applied their wartime experience to computer design. The main lines of development, as history unfolded, thus go back to Aiken and his Mark I, and especially to Eckert and Mauchly and ENIAC.

Aiken

Howard Aiken was a giant of a man, in body as well as in mind, towering six feet four inches. A man of deep convictions, he had strong likes and dislikes. He often formed his opinion of others at the instant of first encounter. It was said that on a scale from 1 to 10 he rated people as either 0 or 11 — there was no middle ground with people, as there was none in any aspect of his career. Only such a man could have made a reluctant Harvard become a center for the new science and art of computing. With all of his drive and imagination, however, Aiken could not achieve his ends alone. Harvard's computer, or giant calculator, was Aiken's brainchild, but to bring it into being he had to effect the collaboration and cooperation of Harvard, the International Business Machines Company and, eventually, the U.S. Navy. This combination of not always compatible actors, the special technological considerations involved, the contingencies or world war, and the personality of Aiken combined to make the process anything but trouble-free.

Although Aiken achieved world fame as a computer pioneer, he had no idea that he would devote his career to computing or even to applied mathematics when he entered Harvard's Graduate School of Arts and Sciences in 1933 as a candidate for the Ph.D. in physics. He was then 32 years of age, older than most graduate students. He had obtained his undergraduate degree in electrical engineering from the University of Wisconsin and had worked as a power engineer before coming to Harvard.

During Aiken's initial years as a Harvard graduate student, he followed the usual program of studies; he then shifted his allegiance to the field of electronics, the physics of vacuum tubes and the properties of circuits. Aiken's serious interest in machine calculation can be traced back to his early graduate days.

In the course of an oral-history interview that Henry Tropp and I conducted with Aiken, shortly before his death in 1973, he explained that the area of his thesis was "space charge" and that "this is a field where one runs into cylindrical coordinates, or in a parallel case, into ordinary differential equations — in nonlinear terms, of course." "In actual fact" he continued, "the object of the thesis almost became solving nonlinear [differential] equations: not completely, but there was some of that in it." The only methods available in those days were methods of hand calculating and "they were extremely time consuming." So it became apparent "at once, that this could be mechanized and programmed and that an individual didn't have to do this."

A former graduate student, James Hooper, recalls that what Aiken was proudest of having done in his doctoral thesis was to have been able to find a "closed form solution to a partial differential equation." This solution was worked out in terms of Bessel functions, so he "had to do an awful lot of laborious work on a Marchant calculator, interpolating values found in the tables." So we may understand why Aiken decided "to construct a machine to work out the most detailed tables of all the Bessel and related functions that anyone could ever possibly want."

Before long Aiken had gone well beyond the immediate needs of his thesis problem and had begun to think about large-scale calculation by machine. By at least April 1937 Aiken had progressed sufficiently far in his general thinking and design to be ready to seek support from industry. Knowing Aiken's work habits, it is not difficult to imagine that he would have drawn up a careful memorandum stating the features of a proposed machine, its mode of operation, and its general method of solving problems. His philosophy was later expressed in a student's assignment that was drawn up for one of Aiken's classes — the design of an inexpensive laboratory computer (or calculator): "The 'design' of a [...] computing machine is understood to consist in the outlining of its general specifications and the carrying through of a rational determination of its functions, but does not include the actual engineering design of component units."

Aiken thus assumed that the design of a computing machine includes the specification of the logic or the sequence of controlled operations that the machine will be programmed to perform. To judge from the information available, Aiken's design would not have necessarily specified which particular components 'nor even what sorts of components' would be used. The design could apply equally to a machine that would be constructed of mechanical, electromagnetic, or electronic components.

Once Aiken had his plans worked up, he proceeded to seek industrial support for the construction of his machine. The first company he approached was the Monroe Calculating Machine Company, one of America's foremost manufacturers of desk calculators. On 22 April 1937 Aiken had an interview with George C. Chase, a distinguished inventor in the calculator field, who was then Monroe's director of research. Chase has recorded how Aiken outlined his conception and "explained what it could accomplish in the fields of mathematics, science, and sociology." The plan he outlined, according to Chase, "was not restricted to any specific type of mechanism; it embraced a broad coordination of components that could be resolved by various constructive mediums." This accords well with Aiken's philosophy embodied in the earlier quotation from a student's assignment.

It is well known that Aiken's early machines were equipped with electromagnetic relays and switches. I myself was long puzzled by Aiken's apparent preference for relays over vacuum tubes. Accordingly, in the interview, I was poised to discuss with him his reasons for choosing to build his computing machine with relays rather than with vacuum tubes. As part of my preparation for the interview with Aiken, I reread Chase's discussion of his encounter with Aiken. And I even had a copy of Chase's article in my pocket, as reinforcement for my pressing Aiken on the choice of electro-mechanical machine. Early in the interview I raised the question I had prepared. Since Aiken's thesis was on the physics of vacuum tubes, specifically on space charge in the field of electronics, had he ever considered using electronic systems rather than electro-mechanical system? Why had he not contemplated using vacuum tubes? I confess that I had expected Aiken to frame his reply in terms of his great often-expressed idea: reliability. I will even confess that I had, in part, prepared the question less as a means of obtaining information than as an opportunity to record on tape — directly from Aiken's mouth — his thundering condemnation of supposedly unreliable vacuum tubes and his preference for slower and more reliable relays. So you may imagine my astonishment when he replied that he had not been wedded to any particular technology. He had been aware that the construction of his proposed computer would require "money and a lot of it." Since he was not then, nor was he ever, primarily interested in technological innovation, it had seemed to him that the most sensible course was to "build the first machine out of somebody's existing parts," rather than to have to invent or construct parts on his own. Electro-mechanical relays and step switches were already in wide use, teletype had been developed, and there was punched tape or punched cards for input. "The tape," he said, "was harder to edit and you couldn't sort with it, but nevertheless it would work and it had advantages." These "different techniques — printing telegraph techniques — were," he added, "all grist for my mill." At that time, Aiken said, he was "largely a promoter, trying to find out where to get these pieces so that the machine could be put together."

I was not completely satisfied by Aiken's presentation. Accordingly, a little later in the interview, I returned to the subject — why had he not made use of vacuum tubes? This time I stressed the fact that, as a graduate student Aiken's specialty was vacuum tubes and vacuum tube circuits. I asked him specifically whether some thought hadn't been given to having quenching circuits in Mark I, using vacuum tubes. Aiken replied: "Yes. But your question really is: since I had grown up in 'space charge' in a laboratory like Cruft [at Harvard], why wasn't Mark I an electronic device? Again the answer is money. It was going to take a lot of money. Thousands and thousands of parts! It was clear that this thing could be done with electronic parts, too, using the techniques of the digital counters that had been made with vacuum tubes, just a few years before I started, for counting cosmic rays." And than he concluded with the following dramatic assertion: "But what it comes down to is this: if Monroe had decided to pay the bill, this thing would have been made out of mechanical parts. If RCA had been interested, it might have been electronic. And it was made out of tabulating machine parts because IBM was willing to pay the bill."

Before leaving the subject of relays, there is another myth about Aiken that ought to be expunged from the record. This involves the topic of relays and reliability. Aiken's fierce insistence on the highest standards of reliability is well known. What is not so well known is that the original Mark I/ASCC, as it came to Harvard from IBM's assembly system at Endicott, was very unreliable. The worst feature of this unreliability was that it seemed to occur in a random or sporadic way. Eventually Bob Campbell and the operating staff traced the source of the unreliability to the relays supplied to the Mark I from IBM's off-the-shelf inventory.

Designed for the relatively light load of IBM tabulators and cumulators, these relays simply could not stand up under the 24-hour-a-day 7-days-a-week assignment of the Mark I and had to be replaced by heavy-duty components. When Aiken planned the successor Mark II, he asked the Autocall Company to design and manufacture wholly new types of relays, to take the place of those which had not lived up to his standards for the Mark I.

To return to the chronology. When Monroe decided not to construct Aiken's dream machine, he went to IBM. His proposal to them, dated 1937, fills 23 double spaced typed pages. The central portion discusses punched card machines of the IBM type in terms of four design features that are different for punched card accounting "machinery" and "calculating machinery as required in the sciences." These are:

(1) A machine intended for mathematics must "be able to handle both positive and negative quantities," whereas accounting machinery is designed "almost entirely" for "problems of positive numbers."

(2) Calculating machinery for mathematical purposes must "be able to supply and utilize" many kinds of transcendental functions (e.g. trigonometric functions), elliptic, Bessel, and probability functions.

(3) For mathematics, a calculating machine should "be fully automatic in its operation once a process is established." In calculating the value of a function in its expansion in a series, the evaluation of a formula, or numerical integration (in solution of a differential equation), the process, once established, continues "indefinitely until the range of the independent variables is covered" — usually "by successive equal steps."

(4) Calculating machinery designed for mathematics "should be capable of computing lines instead of columns," since very often, in the numerical solution of a differential equation, the computation of a value will be found to depend on preceding values. This is actually "the reverse" of the way in which "existing calculating machines" are capable of evaluating a function by steps.

Aiken concluded this section with the bold statement that these four features are "all that are required" to convert existing punched-card calculating machines ("such as those manufactured by the International Business Machines Company") into machines "specially adapted for scientific purposes." This statement minimized the engineering and design problems that would have to be solved in order to produce Aiken's proposed machine. We cannot tell whether this simple optimism was a reflection of his technological innocence or whether he purposely was making the practical problem of design and construction seem simple in order to convince the executives of IBM that his project as feasible — a gigantic conversion of existing commercial elements rather than the production of something rather essentially new.

In the event, this distinction was to be the central issue in the strong divergence of opinion as to whether Aiken or the IBM engineers should receive primary credit for the invention. IBM eventually produced the machine to fulfill Aiken's expectations. It embodied many new IBM innovations, such as ways of dividing. In the years to come there was considerable acrimony between IBM and Aiken on the question of who had invented the machine — Aiken or the IBM engineers? In retrospect, however, that quarrel has only an antiquarian interest.

But two observations may be made. For IBM this was a one of a kind machine — no one at IBM ever conceived that the ASCC would inauguarate a new product line. The ASCC was to be IBM's contribution to science and its only profit was to be the good publicity resulting from this noble gesture. Aiken, however, saw the machine as only a beginning, the first in a line of machines that would be increasingly powerful and more versatile. For IBM this machine was (and still is) always referred to as the ASCC, the Automatic Sequenced Controlled Calculator; but for Aiken it soon became the Harvard Mark I — the first of a series.

Mark I and its operation

Mark I was enormous, about 20 meters long, 2.5 meters high and almost a meter in width. It was made up of 22 panels, all in a line, plus two supplementary panels jutting out at 90°. Three additional panels, containing a subsidiary mechanism, where added later. Panels 1 and 2 contained 60 constant registers for the input of problems. Each register was controlled by a row of 24 ten-position switches, so that an operator could set by hand as many as 60 23-place constants (23, not 24, since one position was reserved for 0 or 9 to indicate positive or negative numbers). Panels 3 to 11 contained 72 storage registers, each of which also was composed of 24 ten-position switches and was the memory. Three tape readers could feed in values needed for special operations, while another read the instructions. Panels 16 and 17 contained 2 punched card readers and a card punch plus 2 electric typewriters. Thus instructions could take the form of punched tape or punched cards and the output could be type-written tables or punched cards.

Mark I thus embodied a 20th-century variant of an important innovation in making tables that had been suggested by Charles Babbage. The latter, aware that errors were introduced whenever tables were copied by hand (as in typesetting and proof-reading), designed his analytical engine to stamp out the results on a papier maché moulage, from which a stereotype plate could then be cast. Mark I used two IBM electric typewriters to type out its results in tabular form on sheets which were then placed in order, photographed, and transferred to special zinc plates for printing. Aiken used to complain that the operation of the typewriters was lower than the computation and set the maximum speed of production of tables. It may be noted that the operator could watch the numbers being types out as a visual check on the accuracy of the calculation.

In order to see how Mark I was programmed, a few words must first be said about the actual mechanism. Every register had 24 rotating counters, each of which had ten positions, numbered from 0 to 9. Each of these 10-place counters was connected to a shaft that was linked by gears and clutches to central drives that were continually whirring. For every electrical impulse received by a counter, the clutch would engage for an instant, long enough to produce a partial movement equal to a tenth of a complete rotation, advancing the number (or number-position) by 1. The number of such electrical impulses would derive from the instructions on the command tapes. Thus if storage register no. 3 contained the quantity 73,965 (the 19 numbers to the left of the 7 being zeros), and the instruction on the command tape was to add 431, a single pulse would be sent to the counter reading 5, so as advance it one position to 6; three pulses would be sent to the counter reading 6, so as to advance it three positions to 9; and four pulses would be sent to the counter reading 9, so as advance it four positions to 3, while a carry mechanism would advance the next counter from 3 to 4.

A characteristic of Mark I, remembered by anyone who ever witnessed its operation, was the constant humming or whirring sound of the main drive shafts. This pleasant sound has been described as like that of a giant sewing machine. Aiken sat in an office near Mark I, with his door always open, so that he could hear the constant sound of the machine. I well remember being with him one time when he detected a variation in the sound. He leapt out of his seat, ran to a main switch, and stopped the operation until the fault could be located and corrected.

The program for Mark I was entered on a long segment of a roll of paper tape. Each line of the program—that is, each individual command—occupied a horizontal line with twenty-four spaces. These twenty-four spaces were divided into three groups (or fields) of eight spaces each, designated by the letters *A*, *B*, and *C*. The spaces in each field were numbered from 1 to 8, reading from right to left. The tape-reader had twenty-four brush contacts, in positions corresponding to the twenty-four spaces on each line of the tape. The program was entered on the tape by means of holes punched in the tape in the appropriate spaces. When the punched tape was fed through the tape reader, a surge of electric current would pass through each space where there was a hole, sending the appropriate number of electrical impulses to the designated counter and therefore advancing it by the assigned number. No current would pass through the tape at positions for which there was no punched hole.

Essentially, the counters in each register acted as accumulators, performing stages of successive addition. Hence multiplication could be performed as a function of addition. Subtraction required the use of nines-complements, as in tabulators then in use in business machines. Division presented a problem in the stages of the original design. Aiken originally proposed using the famous Newton-Raphson method but the IBM engineers later found that a different set of operations would reduce the number of machine operations.

I have mentioned that the paper tape directing the operations of Mark I used a single horizontal line for each individual instruction and that each such line of instruction consisted of 24 places divided into three groups of eight, known as the *A*, *B*, and *C* fields. The programmer would assign to each field a numerical value determined from the code book, which gave the code numbers for all the different kinds of operations the machine could perform. For example, such a set of three code numbers could be 21; 7321; 32. That is, the *A*-field would have the numerical value 21; the *B*-field 7321; the *C*-field 32. This would be written out symbolically on a form with three columns

OUT	IN	MISC.
21	7321	32

which would designate the register (OUT) from which a number or quantity was to be taken, the operation (MISC.) to be performed, and the register (IN) into which the result of the operation was to be entered. A key-punch operator would convert each line of a program into a set of corresponding punched holes, as in the diagram

A	B	C
○ ○ ○ ○ ○ ○ ● ●	○ ● ○ ○ ○ ● ● ●	○ ○ ○ ○ ○ ● ● ○
8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1

In this diagram, I have separated the three fields for ease of explanation, but on the tapes used in Mark I there would be an equal distance between each assigned space or place and its neighbor.

The code book for programming Mark I was later published in the *Manual of Operations*. Each program consisted of a sequence of commands or instructions. A coded one-line instruction of the form *A*; *B*; *C* could be read: Take the quantity in register *A*, perform the operation designated by *C*, and put the result in register *B*. A typical line of instruction might be coded

OUT	IN	MISC.
21	7321	

which translates as: Take the quantity in the register with code number 21 (register no. 3) and add it to the quantity in the register with code number 7321 (register no. 71).

Aiken devised an ingenious coding system, which provided code designations for each register or storage counter and for a large variety of operations. The registers or storage counters were assigned the following numbers:

Registers	Code Numbers
1	1
2	2
3	21
4	3
5	31
6	32
7	321
8	4
9	41
....	....
71	7321
72	74

Operations were divided into two classes. One was called the "non-automatic" operations. For these the machine would read a line of instruction, perform the designated command, and then step ahead to the next line of the program and stop—not reading that next line and, accordingly, not performing the designated operation. The other set, called the "automatic" operations, were more like those of the computers we know today. That is, the machine would not stop after performing a single operation, but go on to the next command, read the command and perform what was indicated. This automatic procedure could go on for as many steps as were built into the instruction code. For a non-automatic operation to continue, it was necessary to introduce a seven (7), the coding for reading the next line of instruction and performing the designated operation or operations.

Some of the automatic codes were:

divide	76
multiply	761
logarithm	762
exponential	7621
interpolate	763
sine	7631

and some non-automatic codes were:

Read out negative absolute value from storage counter	1
Read out positive absolute value from storage counter	2
Invert read-out of storage counter	32
Step interpolator I ahead	53
Step interpolator III back	542
Print on typewriter I	752

The first example given above represents a typical line of coding for Mark I:

OUT	IN	MISC.
21	7321	32

The operation being called for is coded as 32: to take the number from the storage counter and "invert" it. That is, the machine is instructed to take the quantity in register 3 (code 21) and by means of the invert relay (code 32) find its complement on 9 and add it to the quantity in register 71 (code 7321). It may noted that an operation coding (in the MISC. column) can be assigned the same number as a register (in the IN or OUT columns) without confusion.

In actual practice, however, the machine would rarely be asked to perform a single line of instruction. Thus, in the case of "non-automatic" operations, a 7 would almost always be added in the *C*-field, unless the line was the final one in a sequence.

The first example above would thus become

OUT	IN	MISC.
21	7321	7

which translates as: Take the quantity from register 3 (code 21) and add it to whatever quantity is in register 71 (code 7321); go on to next line of code, read the instruction and proceed to carry out the designated operation. But in the next example, a 7 is not needed, because the designed operation is automatic:

OUT	IN	MISC.
21	7321	32

which (in full detail) translates as: Take x from register 3 (code 21) and subtract it from y in register 71 (code 7321); go on to next line, read the instruction and carry out the designated operation. (Again it should be noted that the 32 in the MISC column refers to the operation of subtraction and does not designate a register as it would if it were in either the OUT or the IN columns.) It should be noted that when the command was to take a quantity from one register, say no. 3, to another, say no. 72, the content of register no. 3 would either be simply entered into register no. 71 or added to whatever number was already there.

Subtraction was performed on Mark I by way of complement arithmetic. The *Manual of Operations* explained the use of complement arithmetic more or less as follows. The complement of any number $W\ X\ Y\ Z$ can be found by taking the complement on *nine* of each successive digit except the last one on the right, for which the complement on *ten* must be taken. Thus the complement of *ten* of $W\ X\ Y\ Z$ is

$$9-W \quad 9-X \quad 9-Y \quad 10-Z$$

or, to use a numerical example, the complement on *ten* of 7528 is 2472 (because 9-7 = 2, 9-5 = 4, 9-2 = 7, 10-8 = 2). Let us now subtract 7528 from any number, say 38421. Then, since

$$38421 - 7528 = (38421 - 1000) + (1000 - 7528)$$

it follows that

$$38421 - 7528 = 28421 + 2472 = 30893$$

In applying this method to machine operation, the process is simplified by using complements on *nine* and "end-around carry." In this process a 1 is automatically carried from the highest column (left-most place) of a machine to the lowest (right-most). The reason is that the complement on *ten* of any number, say ABCD is greater by one than the complement on *nine.* The complement on *ten* of ABCD is

$$9-A \quad 9-B \quad 9-C \quad 10-D$$

while the complement on *nine* is

$$9-A \quad 9-B \quad 9-C \quad 9-D$$

The 1 from the end-around carry raises $9-A$ to $10-A$.

As an example consider a six-column machine, that is, a machine which operates with six significant digits. The complement on *nine* of 7528 is

$$999999 - 007528 = 992741$$

Subtracting 7528 from 38421 now becomes

$$\begin{array}{r} 038421 \\ +992471 \\ \hline 1\ 030892 \end{array}$$

Since the machine has only six significant places, there is no seventh left-most or highest place in the sum for the 1 and so this 1 is carried around the end to the right-most or lowest place to raise the sum from 030892 to 030893, which is the correct answer. Thus our problem of subtraction becomes

$$038421 - 007528 = 038421 + 992471 + 000001 = 030893$$

where the third term (000001) comes from an end-around carry.

Next let us seen how a machine can be programmed to deal with algebraic sign, that is, with negative as well as positive numbers. Once again, consider the example of a six-digit machine. Suppose that our six-digit machine is asked to add any number greater than or equal to 1, say 007364, to 999999. This operation will always produce a 1 in the "seventh" place, which would be lost if there were not an end-around carry to make use of it. Thus in machine operation, with an end-around carry,

$$\begin{array}{r} 999999 \\ +007364 \\ \hline 1\ 007363 \end{array}$$

This addition may also be written as

$$999999 + 007364 + 000001 = 007364$$

This example shows that for machine purposes, the number 999999 has the properties of zero (so long as there is an end-around carry). In general, any machine with end-around carry, can consider a number composed of a string of nines as a zero.

Because subtraction from zero gives a negative number, for machine purposes the complement on nine of any number may be used for the negative of that number. That is, in the case of a six-place machine, we can form -038421 by subtraction from zero

$$-038421 = 000000 - 038421 = 999999 + 007364 + 000001$$

If we add to the machine a supplementary seventh place, in the highest (or left-most) column, which is restricted so that it can read only zero or nine, then we can deal automatically with positive and negative numbers, or with the problems of algebraic sign, by using the zero as the algebraic sign of a positive number and the nine as a negative sign. In general, in order to deal with negative numbers in an automatic manner, "an n-digit calculating machine must be supplied with $(n+1)$ columns, the highest being reserved for the algebraic sign." On Mark I, the A-field thus had only seven useful places or columns, the eighth being reserved for the zero or nine which served as algebraic sign. The zero or nine would be called for by having a hole punched in the eighth place or not.

To see how the algebraic sign worked in practice, consider two examples from a four-place machine. The problem is to subtract 2361 from 7465 by adding -2361 to 7465. The complement on nine of 2361 is 7638 and a nine in the algebraic sign indicates a negative number. Hence this operation may be written as

$$\begin{array}{r} 9\ 7638 \\ +7465 \\ \hline 10\ 5103 \end{array}$$

where the 1 in the non-existent place provides the end-around carry to raise 5103 to the correct answer of 5104. This result can be read as follows: the answer is $+5103$ because the number in the algebraic sign place or column is a zero.

Now let us attempt to add -2361 to 1465 (or to subtract 2361 from 1465). This time we have

$$\begin{array}{r} 9\ 7638 \\ +1465 \\ \hline 9\ 9103 \end{array}$$

in which there is no 1 for an end-around carry. The result will be negative, since the algebraic sign number is 9. But negative numbers are complements on nine so that 9 9103 reduces to −0896 (just as 9 7638 was formed from −2361 at the start of the problem).

Multiplication required several steps. Let it be desired to multiply some number x by y. The first line of code might read

OUT	IN	MISC.
654	761	7

take the number X in register 56 (with code number 654) and enter it (code 761) as the multiplicand, go to the next line of code, read and carry out instructions.

The second line might be

OUT	IN	MISC.
52	0	7

take the number Y in register 18 (code number 52) and enter it as multiplier.

Finally,

OUT	IN	MISC.
0	431	7

take the product XY and deliver it to register 13 (code number 431); proceed to the next line of code.

In practice, however, this would be written more simply in three lines of code as follows:

OUT	IN	MISC.
654	761	
52		
	431	7

1. *Take quantity in register 56 (code 654) as multiplicand (code 761)*
2. *Take quantity in register 18 (code 52) as multiplier*
3. *Deliver product to register (13) (code 431) and go to next line of code and read instructions*

Note that the first 2 lines of coding do not contain a 7. The reason is that the operation of multiplication (code number 761) contains the "automatic" instruction to proceed. Aiken described this feature by noting that "the multiply code is an automatic continue code and therefore replaces the 7's."

Mark I was a *decimal* machine and had a *fixed* decimal point. There were a number of built-in sequences of operations or functions, including logarithms to the base 10, sines, exponentials to base 10, and interpolations. These were officially designated in the *Manual of Operating* as "built-in subroutines."

THE PERFORMANCE OF THE MACHINE THAT BEGAN THE COMPUTER AGE

Mark I became operative in 1943 and was shipped to Harvard in summer 1944. It was immediately turned over to the Navy and used constantly, almost 24 hours a day, 7 days a week, to produce tables of Bessel and Hankel functions and to solve specific problems. Aiken had a large staff provided by the Navy, including 3 programmers — Robert Campbell, Richard Bloch, and Grace Hopper. The mode of solving problems was to have a mathematician decide what numerical method was best adapted to computation by the machine. Then each of the steps had to be written down and translated line by line into numerically coded statements, plus switch-settings needed for the constantsregisters, and so on. The mathematician or programmer worked with the code book at his or her side. Before long, it was seen that certain parts of the instructions occurred again and again. So the practice began of writing such partial programs into a notebook. The most extensive such collection was assembled by Dick Bloch, who was the primary programmer. Aiken told us that Bloch was so skilled a programmer that he would write out his programs in ink! Grace Hopper informs me that she and others also kept private libraries of partial programs. Years later, such collections of partial programs became known as libraries of subroutines, but their origin goes back to the actual programming practices of those who were working on Mark I. These coded subroutines, or "canned" elements of programs, are to be distinguished from the subroutines built into the machine, such as those for logarithms, exponentials, and trigonometric functions.

As originally conceived, Mark I had no conditional or branching circuits. These were added later. Other later enhancements included an electronic multiply/divide unit, additional storage registers, and an improved interpolator tape unit. Throughout its whole life, Mark I operated with a fixed decimal point. Needless to say, Mark I was a strictly decimal machine.

One of Aiken's students (in 1949-1950), Jack Palmer, remembers his experience in programming (coding) Mark I, which he found to have been very "similar to coding modern stored-program computers." The programmer wrote down sequences of codes for addresses and operations in the appropriate columns of a coding form, following which these codes or their equivalents would be punched into a program tape — "a strikingly similar procedure to that which programmers who programmed the early stored-program computers did when they were programming in machine language." They too "needed to know the codes for the addresses of the words in storage and the codes for the operations to be performed," which they would write down "in a strict instruction format on a coding form" and which would later be punched on cards. In both cases, Mark I and the later stored-program computers, the numerical codes were obtained from a coding book.

There is no doubt in the minds of anyone who programmed Mark I that in many ways it was more like a modern computer than other early machines. Where Mark I differed in a fundamental way from later computers was, therefore, not so much in its slower speed (as compared with electronic machines) as in its initial lack of conditional branching and in the complete separation of data and instructions. This latter feature, more than the choice of electronics over electromagnetic relay systems, was and remained central to Aiken's thinking about computers.

Mark I was an extraordinary machine in many ways. It had a long active life from 1944 until it was dismantled 15 years later in 1959. No other of the early giants operated continuously, as Mark I did, for so long at time, 24 hours a day for 7 days of the week. What is even more significant is the fact that Mark I could run so constantly and so long and be relatively free from the kinds of errors that plagued such machines. I have mentioned that Aiken's key word in all his career was reliability. He would gladly sacrifice speed for reliability, if he had been forced to make a choice between them.

In our interview, Aiken recalled that he always tried to find "an identity or some kind of algorithm in the mathematics so that when you compute the number x, you can subject the number x to mathematical scrutiny to show that it's right." As a simple example, he recalled the computation of trigonometric tables "for our own use." "We wanted $\sin x$," but "we computed $\cos x$ too, and we squared and added them to make sure they were equal to one." Even then Mark I was tabulating results for internal use, that is, for direct storage on tape or cards, the results were also typed so that a visual check could be made. Aiken also noted, 'We had a check counter, and if you subtracted two numbers in that check counter, and the absolute value of the difference was greater than the preassigned value the machine stopped."

In 1944, Aiken was asked to build a new machine for the Navy. Since Mark I functioning well, he did not modify the basic design; accordingly, it too functioned by relays. Mark II was considerably faster than Mark I, in part because the mechanical register system was replaced by a relay-system. Furthermore, the relays themselves were faster. From the point of view of computer history, Mark II was innovative in that it could be split into two separately operative machines that could work independently on smaller problems. Each half 50 storage registers, two multiplication units, 1 addition unit, two separate tape readers for instructions and four more for data. Mark II was decimal, with ten digit numbers, and had a floating decimal point; the digits were stored in a specially devised binary-coded decimal system. The hardwired subroutines were extended to include

$$1/x; \quad 1/sqrt(x); \quad ln(x); \quad exp(x) \quad cos(x) \text{ and } arc\ tan(x).$$

The two halves, furthermore, could be joined in serial or parallel connection.

SPEEDS

	Mark I	Mark II	Mark III	Mark IV
addition	0.3 sec(1u)	0.2 sec	4 msec	3.6 msec
multiplication	up to 6 sec(20u)	0.7 sec	12 msec	12 ms
division	up to 15.6 sec(52u)			26.4 ms
look up built in functions	60-89.4 sec (199u-298u)	5-12 sec		

Mark III was designed in 1949 for the Navy. It differed from Mark I and II in using magnetic drums. A special feature was a mathematical button board, to put in subroutines automatically. The speed of performing multiplication was 12.75 milliseconds. Aiken called it the 'slowest electronic machine' in existence. It had 5000 vacuum tubes and 200 relays.

Mark IV used ferrite magnetic cores. It was built for the Air Force and signalled Aiken's departure from the field of machine building.

How shall credit be apportioned for the pioneering machine — the Mark I/IBM ASCC? Aiken was always aware that the IBM engineers and their associates would never have invented, much less ever have thought of, the ASCC if he hadn't come along with his proposal. In this sense how could he *not* have considered himself the first mover, the instigator, the primary inventor! He was, to use a mathematical expression, the necessary condition. In the sense of having conceived an automatic sequenced calculator, Aiken was the progenitor, the inventor of the idea and the function. But he was also unable — of and by himself — to convert his ideas into a machine; he was not, to use the other part of that mathematical expression, sufficient. He needed IBM or some other company with practical experience in electrical and electro-mechanical devices. From the IBM point of view the ASCC had been put together by IBM engineers, using standard IBM parts and significant new devices invented at IBM and not at Harvard. On the practical side of invention, Aiken's contribution would have appeared minimal, even though his initiative and general proposal had been responsible for the project that produced the machine.

Frederic P. Brooks, Jr. had suggested "a technical reason for some of the misunderstanding" that arose between IBM and Aiken with respect to credit. He notes that "Gerrit Blaauw's distinction between the architecture, the implementation, and the realization of computers was by no means understood at that time." Today "its clear that with the Mark I the architecture, and what we today would call a good bit of the implementation, had been designed by Aiken." And "the rest of the implementation and all of the realization had been designed by Durfee, Lake and Hamilton" at IBM. Brooks concludes that "a distinction between those roles, if it had been clear at the time, would probably have diffused some of the bitter emotional feeling over the question of credit."

The Harvard Mark I, the IBM Automatic Sequence Controlled Calculator, was not a machine that set design standards for an industry, but rather was the first real demonstration that such machines were practicable. It is a fact of historical record that Mark I was the machine that first proved to the world at large that a complex calculating engine could function automatically, performing operations in sequence, and could follow a predetermined program from the entry of the data to the production of the final results. The world-wide publicity attendant on these achievements, aggrandized by the stark fact of regular and continuous operation to produce reliable and accurate results, convinced any last doubters that large-scale automatically sequenced calculators were practical and could perform a major role in our technical world. In this sense, it is certainly correct to say that when the switch on Mark I was thrown the Computer Age began.

AIKEN'S SIGNIFIANCE

Aiken is sometimes held to be reactionary because he was always weary of the concept of the "stored program" and did not incorporate it into any of his later machines. This stance *did* put him out of step with the main lines of computer architecture in what we may call the post-Aiken era. It must be kept in mind, however, that there are vast fields of computer application today in which separate identity of programs and data must be maintained, for example, in telephone technology and in what is known as ROM ("read-only memory"). In fact, computers without the stored-program feature are often designated today (for instance, by Texas Instruments Corporation) as embodying what is called "Harvard architecture," by which is meant "Aiken architecture."

Howard Aiken's place in the history of computers, however, cannot be measured by his four machines, important as they may have been. He recognized from the start that the computers being planned and constructed would require mathematicians to program them, and he was aware of the shortage of such mathematically trained men and women. To fill this need Aiken convinced Harvard —against its will— to establish a course of studies leading to the master's degree and eventually also the doctorate in what has become known as computer science. Just as Aiken —by the force of his success, abetted by his ability to find outside funding for his programs— achieved tenure and rose to become the first full professor in the new domain of computer science, so he inaugurated at Harvard what appears to have been the first such academic program to be put in place anywhere in the world. The roster of his students contains the names of many who became well known in this subject, including Gerrit Blaauw, Frederick Brooks junior, Kenneth Iverson, and Anthony Oettinger. As other academic programs came into being, they drew directly or indirectly on Aiken's experience at Harvard.

A third area of archievement, one of real significance, was his organization at Harvard, in the post-World War II years, of a series of international conferences which brought together almost everyone of any significance in the general area of the new science and art of computers. This was part of his general course of action to forward the design, manufacture, and use of computers.

A fourth area was his constant search of new applications for computers and ways of enticing scholars to use the new machines. For example he sought out his colleague in the Economics Department, Wassily Leontief, and got him to use the Mark II for his input-output economics. He lectured extensively in Europe and in America on Computers and their possible applications. As an enthusiast he always sought out and welcomed new fields of applications for computers and encouraged his students and others to be on the constant lookout for new applications that would enhance the potentialities of this new instrument. He pioneered the new world of application of computers to business at a time when the computer world was almost exclusively concentrated on problems arising in science and engineering or in government (national security and the military). Aiken's research under contracts with the American Gas Institute and the Bell Telephone Laboratories inaugurated the present system of computer billing.

Fred Brooks has summarized this part of Aiken's activites as follows:

> "He was one of the very first to realize the important potential of computers for business, and that business applications would completely dominate scientific applications. He insisted that the business applications would require usability of decimal among other things, and he turned his attention to forging the ties with the utilities, the business organizations, that would first have the need — insurance companies, and that kind... — in order to make sure that the mathematical approach was carried over into the business problem."

Aiken was a visionary, a man apt to be ahead of his times. Grace Hopper and others remember his prediction in the late 1940s, even before the vacuum tube had been wholly replaced by the transistor, that the time would come when a machine even more powerful than the giant machines of those days could be fitted into a space as small as a shoe-box. His students and associates did not know whether to take him seriously. Toward the end of our interview, just weeks before his death in 1973, Aiken made another prediction. We were talking about how the cost of computing power had been constantly and rapidly decreasing. Aiken pointed out that hardware considerations alone did not give a true picture of computer costs. As hardware has become cheaper, software has become more expensive. And then he gave us his final prediction: "The day will come," he said, "when manufactures will give away hardware in order to sell software." Time alone will tell whether or not his final look-ahead into the future was correct.

STORAGE COUNTERS

No.	Code	No.	Code	No.	Code
1	1	25	541	49	651
2	2	26	542	50	652
3	21	27	5421	51	6521
4	3	28	543	52	653
5	31	29	5431	53	6531
6	32	30	5432	54	6532
7	321	31	5431	55	65321
8	4	32	6	56	654
9	41	33	61	57	6541
10	42	34	62	58	6542
11	421	35	621	59	65421
12	43	36	63	60	6543
13	431	37	631	61	65431
14	432	38	632	62	65432
15	4321	39	6321	63	654321
16	5	40	64	64	7
17	51	41	641	65	71
18	52	42	642	66	72
19	521	43	6421	67	721
20	53	44	643	68	73
21	531	45	6431	69	731
22	532	46	6432	70	732
23	5321	47	64321	71	7321
24	54	48	65	72	74

SWITCHES

No.	Code	No.	Code	No.	Code
1	741	21	75431	41	7651
2	742	22	75432	42	7652
3	7421	23	754321	43	76521
4	743	24	76	44	7653
5	7431	25	761	45	76531
6	7432	26	762	46	76532
7	74321	27	7621	47	765321
8	75	28	763	48	7654
9	751	29	7631	49	76541
10	752	30	7632	50	76542
11	7521	31	76321	51	765421
12	753	32	764	52	76543
13	7531	33	7641	53	765431
14	7532	34	7642	54	765432
15	75321	35	76421	55	7654321
16	754	36	7643	56	8
17	7541	37	76431	57	81
18	7542	38	76432	58	82
19	75421	39	764321	59	821
20	7543	40	765	60	83