

Engineering “The Miracle of the ENIAC”: Implementing the Modern Code Paradigm

Thomas Haigh
University of Wisconsin–Milwaukee

Mark Priestley
Crispin Rope

Building on plans developed by Adele Goldstine and others, in April 1948, Nick Metropolis completed the conversion of ENIAC to the style of programming first described in John von Neumann’s 1947 “First Draft of a Report on the EDVAC,” making ENIAC the first computer to execute programs written in the new style. This article documents the conversion process and compares the 1948 ENIAC’s capabilities to those of the first modern computers.

The Electronic Numerical Integrator and Computer, or ENIAC, created at the University of Pennsylvania between 1943 and 1945, has not been neglected by historians of computing. After receiving considerable publicity when first unveiled in 1946, it was at the center of a series of high-profile patent lawsuits in the 1960s and 1970s. Its 50th anniversary was marked by the university with lavish celebrations presided over by the sitting vice president of the United States and a reunion organized by the US Army, on whose behalf it was constructed.¹ ENIAC has been reproduced on a chip and emulated in software.² Several scholars have recently attempted to reconstruct early ENIAC programs, learning its unique system of switches and wires.³ In recent years, its dwindling band of female programmer-operators achieved minor celebrity as female pioneers of computing.⁴ ENIAC occupied a central space in the long-running Smithsonian exhibit on the Information Age as a kind of corridor through which visitors walked to reemerge in the computer age. Its corresponding, and sometimes unearned, centrality to many different narratives about computer history was memorably critiqued by Michael S. Mahoney in his 2005 paper “The Histories of Computing(s).”⁵

Despite this, ENIAC’s post-1945 career as a functioning computer deserves closer

attention. Lawyers and historians have been interested primarily in questions of priority—for example, when did ENIAC first work, was it really a computer, was it truly “general purpose”? Both groups thus focused on events from the beginning of ENIAC’s design in 1943 to its initial operation at the Moore School in late 1945 and 1946. For lawyers, the date on which the ENIAC patent was filed delineated a time span of evidence admissible to challenge or defend its validity. Historians focused similarly because we often care more about what ENIAC led to, the stored-program computer, than what it actually did.

Analysis of ENIAC as an innovation has therefore tended to focus on the process by which the machine was initially designed, primarily by J. Presper Eckert and John Mauchly, and built at the University of Pennsylvania’s Moore School. In most accounts, including Atsushi Akera’s award-winning *Computing a Natural World* and Scott McCartney’s journalistic *ENIAC*, the story switches elsewhere once the machine is operational and its creators disperse to new projects.⁶ We know a lot about ENIAC’s conception and birth at the Moore School, but its productive adulthood and doddering old age were spent at the Army Ballistics Research Laboratory (BRL) and have been little explored by scholarly historians.⁷

In this article, we focus on the period between mid-1947 and late 1948 as the most transformative period in ENIAC’s later life. It entered this period a sulky teenager, traumatized by its move to a new town and burdened with a programming system that made it hard for even its closest friends to persuade it to help them. By May 1948, what renowned Los Alamos mathematician Stanislaw Ulam called the “miracle of the ENIAC” had occurred after it was converted to a new control system.⁸ ENIAC finished 1948 triumphant, as the world’s fastest and easiest-to-program computer and a lynchpin of the American nuclear weapons program.

The most important change was the design and implementation of an entirely new programming system, based on principles first described by John von Neumann in 1945 in the seminal “First Draft of a Report on the EDVAC” (hereafter simply “the First Draft”). This provided ENIAC with the control method planned for the new breed of computers then under construction. We refer to this, here and in the first article in this series, as the “modern code paradigm.” ENIAC was the first computer to execute a program written in this modern form, a feat that has been little recognized by historians.

Those unexcited by talk of “firsts” may nevertheless find ENIAC’s transformation a striking demonstration of what is sometimes called “user-driven innovation.”⁹ Scholars are aware that the many little tweaks and improvements needed to reshape new inventions into transformative innovations are often performed by the people and organizations adopting the new technology. However, historians have tended to treat ENIAC as a largely static artifact, with capabilities and a historical significance that were fixed at the Moore School by 1945. From the viewpoint of its later programmers, ENIAC as used from 1948 onward was more fundamentally a creation of von Neumann’s collaborators in Princeton (with considerable assistance from the computing staff at the Ballistics Research Laboratory and a group of contractors led by Betty Jean Bartik) than of its original designers Eckert and Mauchly.

The first article in this series, “Reconsidering the Stored Program Concept,” examined the history of the aforesaid idea and proposed a set of more specific alternatives.¹⁰ The third, “Los Alamos Bets on ENIAC: Nuclear Monte Carlo Simulations, 1947–1948” (to appear in *IEEE Annals*, vol. 36, no. 3, 2014), will examine in detail the first

program run on the machine after its conversion to the new programming method.

Contexts

By 1946 several projects were already underway to build computers based on the ideas described in von Neumann’s First Draft, following a quite different design from that used for ENIAC. That year, computing pioneers from across the United States (plus a few from England) convened at the Moore School to learn how to build their own machines. Still the world’s most powerful computer, ENIAC was temporarily housed in the same building. Yet the event’s organizers, among them ENIAC’s own creators, had not originally planned to give students access to the machine, so sure were they that its approach had already been superseded.

Not long afterward, ENIAC was finally taken to pieces and delivered to Aberdeen. Getting it back into a usable state was a lengthy process. A log entry dated 29 July 1947 states that “ENIAC starts up again,” and this has generally been taken as the date on which it resumed work, although its recovery was slow enough that little useful work was performed that year.¹¹ Even before ENIAC was restarted in its new home, plans were being drawn up to fundamentally modify its programming style.

ENIAC’s architecture might already have begun to seem outmoded to computer designers, but as the fastest machine then in existence, it was much more attractive to groups with massive computations to perform. Those making the “pilgrimage” to Aberdeen included John von Neumann.¹² As the only credited contributor to the First Draft, he was recognized as an intellectual force behind the new breed of computers, and as a faculty member at the Institute of Advanced Studies (IAS), he headed an influential early project to build an EDVAC-type computer and to research the new coding and analysis methods needed to make use of such a machine.¹³

Los Alamos had a backlog of large-scale computing needs, to which the institute’s machine would be applied as soon as it became operational. In early 1947, that point remained several years away, even by the most optimistic estimates. Thus, when von Neumann, with Los Alamos scientists Robert Richtmyer and Stanislaw Ulam, decided to apply the new concept of Monte Carlo simulation to nuclear chain reactions, he knew

that ENIAC offered the best chance of applying it in the near future. By March he had charged his close collaborators Herman and Adele Goldstine, both veterans of ENIAC's Moore School days, to investigate the "set-up" of ENIAC for Monte Carlo.¹⁴

Launching the Conversion Project

ENIAC control was originally highly distributed. It was configured to run a particular program by wiring problem-specific connections between several dozen constituent units and configuring a mass of switches spread over those units. W. Barkley Fritz, a member of the ENIAC team at the BRL, later wrote that the process "can best be described as analogous to the design and development of a special-purpose computer out of ENIAC component parts for each new application."¹⁵

As a result, constraints on the programming capacity of ENIAC were spread across many aspects of its design. These various constraints interacted in unpredictable ways to further constrain the logical complexity of the program. ENIAC programmers set up individual instructions on the program controls belonging to the machine's various units. The number and type of these controls varied from unit to unit. For example, 240 controls were spread across the 20 accumulators, providing for numerical transfers, additions, and subtractions, while each of the 24 controls on the high-speed multiplier defined a single multiplication operation. The number of controls placed an upper limit on the number of instructions each type a program could contain. The master programmer contained only 10 steppers, which further limited the overall logical complexity of a program. Setting up a conditional test not only tied up one of the master programmer's steppers but also monopolized one output terminal of an accumulator, which limited the usefulness of that accumulator for other purposes. As the machine's designers commented in 1945, "in planning a set-up of a problem for the ENIAC the inner economy of the machine must be considered in allocation [sic] program facilities to various parts of the program."¹⁶

Around April 1947, John von Neumann realized that ENIAC's kit of parts could also be used to build a general-purpose computer controlled by a much simpler and more flexible programming method, similar to that used to control the machine he was then building at the IAS in Princeton. He and his

collaborators were by this point deeply immersed in thinking through the design of their new computer and the programming techniques it would require, producing a series of reports proposing a programming methodology, a rich flow diagramming technique, and methods for handling what they were already beginning to call subroutines.¹⁷ At that point, they had probably thought more deeply about these issues than any other group in the world, so it is not entirely surprising that they applied the same concepts to their work on ENIAC.

Its switches and wires would be configured to decode and execute a fixed repertoire of basic instructions. ENIAC was equipped with three "function tables," read-only memories studded with large numbers of knobs. Each knob set a single decimal digit. These had originally been intended to store numerical constants and lookup tables for function values. Because they were relatively plentiful and gave high-speed, random access to the numbers stored in them, the function tables provided an excellent location for program storage.

Work on the Monte Carlo simulation program for Los Alamos took place in parallel with the design of the new instruction set. Von Neumann's group in Princeton included his acolyte Herman Goldstine. Their wives Klara von Neumann and Adele Goldstine were not IAS staff members but were interested in the new work. Los Alamos hired Adele Goldstine as a consultant to assist with the project, effective 7 June 1947.¹⁸ Klara von Neumann was also hired around the same time.¹⁹ Although young, still just 26 years old, Goldstine's qualifications were unmatched. She had degrees in mathematics from Hunter College and the University of Michigan and had joined the staff of the Moore School in 1943, the year after arriving in Philadelphia with her husband Herman. She had been involved in planning for ENIAC programming even before the machine was finished and had trained the human "computers" at the Moore School, among them many of ENIAC's first programmers, in mathematical techniques. In 1946 she had compiled two volumes of *A Report on the ENIAC*, somewhat confusingly called "Part 1 Technical Description of the ENIAC." They remain the most detailed guide ever produced to the functioning and programming of ENIAC in its original mode.

Thus, the team combined an unmatched knowledge of the new programming concepts with an unparalleled understanding of

**By mid-May 1947,
intensive planning work
was underway for
ENIAC’s conversion.**

ENIAC’s capabilities. In fact, Goldstine had already written in some detail about the possibility of using function tables to generate control pulses in her 1946 ENIAC documentation, the relevant part of which ultimately found its way into the ENIAC patent.²⁰ This provided additional flexibility in sequencing the initiation of up to 14 small programs, each configured in the traditional manner. The 1947 approach was quite different, storing programs as numerical data rather than directly generating control pulses from the function tables. Still, Goldstine had already documented the tables as control devices, so this perhaps made the next leap seem more natural to her collaborators.

Richard Clippinger, who was at that time a mathematician working at BRL, later claimed credit for the initial idea of converting ENIAC to an EDVAC-like control method.²¹ However, in a 1948 report he had presented a different account:

In the Spring of 1947, J[ohn] von Neumann suggested to the author that it would be possible to run the ENIAC in a way very different from the way contemplated when it was designed.... his suggestion has been worked into a finished regime by J. von Neumann, [Adele] Goldstine, [Betty Jean] Bartik, [Richard] Clippinger, and [Art] Gehring with contributions by A. Galbraith, [John] Giese, [Kay] McNulty, [John] Holberton, [“Betty”] Snyder [later Holberton], [Ed] Schlain, [Kathe] Jacobi, [Frances] Bilas, and [Sally] Spear. The role of J. von Neumann in working out the details has been a central one.²²

Clippinger’s testimony in the ENIAC trial clarifies the discrepancy. Under questioning, his specific claim was to have originated the earlier concept of sequencing subroutines via function table pulses. Clippinger testified that he had come up with the idea when working on a complex set of airflow calculations and that Goldstine had taken it from him without

credit. The entire subsequent conversion effort was, in his mind, merely “several evolutionary stages” in refinement of his fundamental insight. Clippinger testified to having made his invention only after learning about the dummy program technique from Adele Goldstine in “April or May” of 1946 but was then, under cross examination, unable to explain why an engineering diagram for the necessary “F.T. Program Adapter” had already been drawn up in January of that year.²³

J. Presper Eckert, who led electronics design work on ENIAC, later wrote that when designing ENIAC “we expected that at some time someone would want to do this, so we built the necessary cable.... Clippinger later ‘rediscovered’ these uses of the function tables, without knowing that they had already been provided for in the original hardware.”²⁴ Clippinger may well have contributed to some planned elaboration of the technique to generate control pulses directly from the function table, which would explain the decision taken by BRL early in 1947 to order a new device known as the “converter.” However, it seems clear that he provided neither the original idea for direct function table control nor the later idea for ENIAC’s conversion to the modern code paradigm.

The 51 Order Code

By mid-May 1947, intensive planning work was underway for ENIAC’s conversion. Clippinger had previously arranged for Jean Bartik, a member of the original team of ENIAC programmer-operators who did not move to Aberdeen with the machine, to lead a five-person group in Philadelphia doing work under a one year contract to BRL.²⁵ Conversion planning was its largest project. Bartik recalls a delegation from BRL traveling up to Princeton with Clippinger for several visits of “two or three days” to work with Adele Goldstine, during which John von Neumann would spend “maybe half an hour a day” with them.²⁶

A letter written by John von Neumann to R.H. Kent at BRL stated, “Four weeks have passed since I started to discuss with you, and in detail with Clippinger, the new method of setting-up and programming ENIAC, and two weeks since we concluded these preliminary discussions.” The letter was sent on 13 June 1947, so detailed discussion with Clippinger began in mid-May and continued intensively until the end of that month. It also mentioned that “several variants of this method have been worked out by Clippinger

and his group, on the one hand, and the Goldstines, on the other.²⁷ This led to what were known informally as the Princeton configuration, from the Goldstines, and the Aberdeen configuration, from Bartik and Clippinger.

The Philadelphia and Princeton teams met regularly through the summer and autumn of 1947. Adele Goldstine made several collaborative trips to Philadelphia and at least one to Aberdeen.²⁸ By July, she had written up a detailed preliminary conversion plan and instruction set. Her report called this a method of “central control,” which nicely captures its consolidation of all application-specific programming in the function table switches.²⁹ Conversion plans included instruction sets and “set up” diagrams for ENIAC showing the wiring and switch configurations needed to decode and execute the instructions.

Those involved tended to refer to the various plans by the number of instruction types (“orders” in the parlance of the day) included; hence this initial plan was the “51 order code.” Each instruction was identified by a unique two-digit code. It required no modifications or additions to ENIAC’s original hardware, meaning that the number of instruction codes was constrained by the capacity of the master programmer unit.

The basic principles documented by Adele Goldstine in July 1947 remained fundamental to the central coding system used for the rest of ENIAC’s career. Building an EDVAC-style computer out of ENIAC’s “kit of parts” meant designing a machine architecture as well as an instruction set. To do this, the team made extensive use of the principles developed in von Neumann’s First Draft and subsequent work, in particular the fundamental distinction between memory, arithmetic, and control units. Space here does not permit a full discussion of ENIAC’s instruction set, new programming method, or the details of its method of conversion. These have been discussed previously by a number of authors.³⁰

After conversion, ENIAC stored programs and data interchangeably in an addressable read-only memory. Approximately 3,600 digits available across three function tables were available for parameters, tables of constants, and instructions. These function tables consisted of banks of rotary switches, each with positions 0–9. ENIAC programs could read data values or instructions from any part of the function tables but could not modify them.

The basic principles documented by Adele Goldstine in July 1947 remained fundamental to the central coding system used for the rest of ENIAC’s career.

ENIAC’s basic structure might be likened to a platoon of largely autonomous electronic adding machines marching to the common beat of its cycling unit. These accumulators exchanged control pulses with each other and with ENIAC’s other units, which included specialized arithmetic units, input/output units, and a “master programmer” that counted off loop cycles. These signals triggered whatever actions had been set up by configuring a set of dedicated switches and dials known as “program controls” to occur when a pulse arrived on a wire attached to a particular input terminal. Thus, the control pulses (or, as they were often called, “program pulses”) used to coordinate units did not directly encode arguments or instructions but sent just one message: “Go!” The actions initiated often involved sending or receiving data, but this took place using different sets of terminals and different kinds of cables.

It was not easy to centralize the control of such a computer to support the modern code paradigm because the latter was designed around a fundamentally different architecture. Goldstine’s draft thus imposed the virtual outline of the abstract computer discussed in the 1945 First Draft over the complexity of ENIAC. It partitioned the 20 accumulators into three functional groups, modeled on what today is often referred to as the von Neumann architecture.

This provided the basic structure needed to fetch and decode instructions. Instructions were identified by a two-digit code, and up to six instructions could be stored in each line of the read-only function tables. After conversion, ENIAC offered programmers capabilities

fundamentally similar to those of the planned IAS computer. Each instruction was implemented through “local” programming using the traditional system of switches and wires to trigger basic ENIAC operations, such as clearing or incrementing accumulators and transferring numbers between units. This is reminiscent of the later practice of microprogramming. For example, the FTN instruction, a mnemonic for Function Table Numerics, was described in later documentation as a combination of six distinct actions. One was incrementing the address stored in the first three digits of accumulator 8, while another was clearing accumulator 11.³¹

When Goldstine partitioned ENIAC’s hardware into what von Neumann had called “organs,” she assigned eight accumulators to the “arithmetic system.” Accumulator 15 became the machine’s “accumulator” in the modern sense. She called it a “central arithmetic and transfer organ.” Accumulator 13 was an auxiliary arithmetic register. The six other accumulators placed within the arithmetic system were akin to special-purpose registers in later computer architectures and were attached to ENIAC’s specialized hardware units to serve as fixed sources or destinations for the arguments and results of multiplication, division and square root operations. Accumulators 11 and 15 were also used as buffers to hold data read from the function table or from punched cards. The arithmetic operations provided were based on those John von Neumann’s team was then refining for its computer at the IAS, with a few modifications made necessary by the properties of ENIAC’s original arithmetic circuits.³²

The new architecture allocated three accumulators to a “control system.” This initially included what would later be called an instruction register, although hardware changes made that unnecessary in later revisions of the plan. The control system also held several four-digit function table addresses. Von Neumann’s EDVAC design had relied on address modification for several common operations, such as conditional jumps and data reads from calculated locations.¹⁰ When planning ENIAC’s conversion, Goldstine and her colleagues wanted to preserve the capability of specifying at runtime the address on which an instruction should operate. Because the program instructions themselves could not be modified, they instead designed several ENIAC instructions to work on addresses stored in control system

accumulators, where they could be manipulated programmatically.

One was the “current control argument,” the address of the current line of instructions. By default, this was incremented each time the instructions in a line were completed so that instructions were executed sequentially. An “unconditional transfer” instruction was provided that overwrote the current control argument with a new address, thus changing the location from which the next instruction code would be read. The other stored address, the “future control argument,” was set with a “substitution order.” When a conditional transfer instruction was executed, if the sign of the number in accumulator 15 was positive, the transfer was carried out by overwriting the current control argument with the future control argument. As well as conditional transfers, this technique was used to implement subroutine returns and the “variable remote connections” used in von Neumann and (Herman) Goldstine’s flow diagram notation to jump to a location determined and stored earlier in the computation.³³ In this case, the limitations of ENIAC’s hardware inspired a new mechanism that directly paralleled the abstractions used in the flow diagrams. These stored control addresses gave ENIAC a simple form of indirect addressing. More elegant indirect addressing mechanisms, such as index registers, were widely relied on by later computers to limit the need for code modification.

In this initial plan, nine accumulators remained after other needs were addressed. The arithmetic and programming capabilities of these accumulators were hidden so that they appeared to the programmer as simple storage devices. Together with a much larger expanse of read-only memory on the function tables, these accumulators constituted what von Neumann had called the “memory organ.” A pair of orders was defined for each accumulator: a “talk” order that copied its data into accumulator 15 and a “listen” order that did the opposite.

Planning the 60 Order Code

The conversion plans went through several revisions. During the second half of 1947, efforts focused on the “60 order code.” Adele Goldstine was by then less involved with conversion planning because her focus had shifted to programming a different kind of nuclear simulation, codenamed Hippo, in collaboration with Robert Richtmyer. Activity shifted to the ENIAC team at Aberdeen

and to Bartik's group of subcontractors in Philadelphia.³⁴

At this point, a new piece of hardware appeared in the plans. Connecting an additional 10-position electronic "stepper" unit to ENIAC's master programmer unit would speed the decoding of the first digit of each instruction. Based on this initial digit, a signal would be sent to one of the 10 internal steppers, originally used to control nested loops. The internal steppers had six "stages," each with its own output signal, and could thus respond to values of the second digit in the range 0 to 5 by firing off a control pulse on one of 60 output wires leading from the master programmer to the rest of ENIAC. This redesign simplified the planned control system and freed up an additional register for general-purpose storage.³⁵

The various 60 order code designs built around this approach maintained the basic structure of Goldstine's original, diverging principally by incorporating more control instructions (to set up the current and future control addresses and the function table argument) and a much more flexible set of shift instructions. By 13 November 1947, Clippinger had summarized the work of Bartik's group into a proposed 60-instruction set.³⁶ One week later, a 19-page treatment of the 60 order code was issued, including full instruction definitions and detailed plans for setting up ENIAC to execute the code, including pulse timings, pulse amplifier configurations, and settings for the individual units.³⁷ It consolidated the Princeton and Aberdeen variant instruction sets into a single document. Around the same time, ENIAC programmer Betty Holberton was working on a suite of test programs for the various units, on which she received feedback from others, including Clippinger and Goldstine.³⁸ None of these plans incorporated a 1,000-digit delay-line memory often called "the register" that BRL had decided back in February 1947 to equip ENIAC with. This presumably reflected a well-grounded pessimism toward the likelihood of it working reliably in time for the initial conversion (or, as it turned out, ever).³⁹

Conversion plans were announced during a press conference at Aberdeen on 12 December 1947 and reported in the *New York Times* the next day. This coincided with Clippinger's initial presentation of the planned conversion to a technical audience at the 1947 ACM meeting.⁴⁰ In a report that repeatedly referred to ENIAC as a "robot," the *Times*

Rather than argue about whether 1948 ENIAC was a stored-program computer, we prefer to define its historical significance in more precise terms.

disclosed that these changes would give it "a substantial part of the efficiency which is being built into the Edvac."⁴¹

Implementation

Nicholas G. "Nick" Metropolis, a Los Alamos veteran and confidant of John von Neumann then working at the University of Chicago's Institute of Nuclear Studies, was scheduled to arrive in Aberdeen to begin preparations for Monte Carlo on 20 February 1948, by which point BRL staff were expected to be "well on the way" to reconfiguring ENIAC following the expected completion of modifications by the Moore School team on 9 February.⁴² In fact, reconfiguration for the 60 order code had not started when he and Klara von Neumann arrived, so this became a short preparatory trip. According to Metropolis, the two then began their own work to design an expanded instruction set after coming across the "converter," a newly arrived piece of hardware, which they realized could be used to efficiently decode all 100 possible two digit codes.⁴³ BRL had committed to installing the converter a year earlier, but it was not used by the 60 order code.⁴⁴ Incorporating the converter expanded the available instruction set, speeded operation, and freed up the master programmer unit. The treatment of shifts was modified significantly, with two parameterized instructions replaced by 20 different instruction codes, each with a fixed meaning.

The log book maintained by ENIAC's operators makes clear that, contrary to several previous accounts, the 60 order code was never installed.⁴⁵ It is unclear if the planned additional stepper was ever fitted, but the master

programmer was not in the end used for instruction decoding.

The converter unit was fitted to ENIAC on 15 March 1948. Two weeks later, on 29 March, the log shows that “Nick [Metropolis] arrives about 4PM and started getting the background coding on for the A.E.C.”—in other words, beginning to configure ENIAC with the new instruction set. The next day, “Nick got the basic sequence & 2 or 3 orders working,” and the day after that ENIAC was “demonstrated for the first time using the new coding scheme.” Various “troubles” slowed progress, and Richard Merwin was summoned from the Moore School to fix the converter. By 6 April, implementation of a “79 order code” was complete.

A lengthy period of testing and debugging followed, full of “intermittents,” “transients,” and other “troubles.” John and Klara von Neumann arrived part way through. John left soon afterward, but Klara stayed behind to work with Metropolis for the rest of his time there. According to the ENIAC Operations Log, a demonstration to students of the Applied Physics Laboratory on 12 April “went off without hitches” and “was the first adequate demonstration using the new coding techniques.” This was thus the first time code written in the modern paradigm had been successfully executed.

The “first production run” on the Monte Carlo problem followed on 17 April, but “troubles of various kinds” returned immediately. By 23 April, morale was flagging, and the log recorded that “no progress [was] made for the day. It looks like a major overhaul or something drastic is necessary to make the Eniac work properly.” Metropolis salvaged the situation on 28 April with a more modest intervention: dialing down ENIAC’s clock from 100 KHz to 60 KHz. Reliable operation suddenly became the norm rather than the exception. Production work on the problem began in earnest, completing on 10 May 1948. Stanislaw Ulam, one of several who had been following the progress of the work with interest, wrote to John von Neumann that “I heard from Nick on the telephone that the miracle of the Eniac really took place.”⁴⁶

After Metropolis and Klara von Neumann left, Clippinger took control of the machine. He tweaked its configuration to implement an 83 order code and revised the “enumeration of lines” imposed by Metropolis to bring them into consistency with earlier plans.⁴⁷ Clippinger was then able to run a

long-planned supersonic air-flow simulation program. By 6 July, specification of a 100 order code was complete, which actually provided just 84 distinct instructions.⁴⁸ Two more of its instructions were implemented on 12 July. After this, the instruction set was essentially unchanged for the next five years. All subsequent ENIAC applications were programmed in the modern code paradigm.⁴⁹ By August 1952, ENIAC had tackled around 75 problems in this mode.⁵⁰

ENIAC as a Stored-Program Computer

Having documented the capabilities of the converted ENIAC (referred to hereafter as 1948 ENIAC, to distinguish it from the original 1945 ENIAC), we are now in a position to analyze it against the first generation of EDVAC-inspired computers completed over the next few years.

Machines based on the EDVAC design are usually called “stored program computers.” The current literature gives a rather contradictory picture regarding the status of ENIAC as a stored-program computer, reflecting the inherent ambiguity of the concept. Those who have looked closely at ENIAC in its post-1948 configuration have generally described it as a stored-program computer but have qualified this with an adjective or two.

In one of the earliest articles on computer history, Nick Metropolis himself suggested that the stored-program computer was not one breakthrough but a series of steps. He credited ENIAC with the third step (“internal control of calculations” in 1946) and the fourth step (“storage control of a computer” in 1948) but suggested that the final step of “read write memories for stored programs” was first implemented in 1949. Thus, one should “credit the ENIAC as the first computer to be run with a read-only stored program” but the “BINAC and the EDSAC as being the first computers to be run with a dynamically modifiable stored program.”⁵¹

This proposed distinction was not taken up by others. Two objections could logically be made to his claim for ENIAC. The first would be that ENIAC was not, even after its modifications, a stored-program computer of any kind. The second would be that it did not become operational in stored-program mode until after a purpose-built stored-program computer, the Manchester Baby, on 21 June 1948.

The second objection, that of timing, is widespread but is unequivocally refuted by

the primary sources we cited earlier. In his influential book, *The Computer from Pascal to von Neumann*, published in 1972 (the same year Metropolis gave his paper), Herman Goldstine devoted several paragraphs to ENIAC's conversion and stated erroneously but with apparent exactitude that "on 16 September 1948 the new system ran on the ENIAC."⁵² That was several months after the Manchester Baby became operational, which appeared to leave little point to debating the status of ENIAC's capability. Later authors, including Hans Neukom, author of the most detailed published treatment of ENIAC's conversion, generally accepted Goldstine's date.⁵³

The first objection, that 1948 ENIAC was not truly a stored-program computer, cannot be dealt with so conclusively because no clear definition of "stored program computer" has ever been agreed on. We have some sympathy with the hyphenated definitions proposed by Metropolis 40 years ago, but time cannot be turned back, and the history of computing community took a different direction. Goldstine himself, who spoke with authority on the topic as a collaborator on von Neumann's seminal papers, called it "a somewhat primitive stored program computer."⁵⁴ In his definitive treatment of von Neumann's computing work, William Aspray described it variously as "a (read only) stored program computer," something "used in stored program mode," and "modified for stored program operation."⁵⁵ However, those same authors elsewhere endorsed EDSAC as the first useful stored-program computer. They thus shared our sense, and that of Metropolis, that ENIAC's conversion implemented key programming ideas and capabilities from the First Draft but that it remained, in some way they did not fully articulate, a crude or partial embodiment of that model.

So, rather than arguing about whether or not 1948 ENIAC was a stored-program computer, we prefer to define its historical significance in more precise terms. We argued in the first article in this series, "Rethinking the Stored-Program Concept," that the complex career of the stored-program concept has hopelessly overloaded it with contradictory meanings. The 1948 ENIAC aligns with some of those meanings but not with others.¹⁰

In that article, we observed that different authors have applied different implicit definitions to the "stored program concept." Some take it narrowly, as the storage of programs and data in a single memory. This has led to claims that IBM's SSEC was the first

stored-program computer or that Eckert and Mauchly had invented the concept by January 1944, before they first met von Neumann. Others define it more broadly, treating it as synonymous with the von Neumann architecture or as encompassing the full scope of the 1945 EDVAC design sketched in the First Draft. It has increasingly been conflated with later conceptions of the computer as an embodiment of the universal Turing machine. Discussion of it as a singular concept has fueled the misperception that there was only one important idea in the First Draft and that this came somehow from Alan Turing.⁵⁶

Thus, we suggested that the concepts embodied in the First Draft and implemented in the dominant computer designs of the 1950s might instead be separated into three distinct clusters. Although unmistakably intertwined, these were in some cases implemented separately in the machines of the late 1940s and have had separate careers over the decades since. First was the EDVAC hardware paradigm, coupling a binary, all-electronic computer to a large, high-speed memory. Second was the von Neumann architecture paradigm, the elegant and minimalist system of central control of serial operations, specialized registers, and division of the computer into separate "organs" for control, arithmetic, and memory. The third cluster formed what we dubbed the modern code paradigm, a set of ideas about the form and capabilities of instructions for such a computer.

We identified six specific features of the modern code paradigm, grounding each one in specific language in the First Draft:

- The program is executed completely automatically, except as needed to prepare input and output devices.
- The program is written as a single sequence of instructions, known as "orders" in the First Draft, which are stored in numbered memory locations along with data.
- Each instruction within the program specifies one of a set of atomic operations made available to the programmer. This usually involved beginning the instruction with one of a small number of operation codes.
- The program's instructions are usually executed in a predetermined sequence.
- However, a program can instruct the computer to depart from this ordinary sequence and jump to a different point in the program.

**After its conversion,
ENIAC’s programming
system embodied the
key features of the
modern code paradigm
as described in the 1945
First Draft.**

- The address on which an instruction acts can change during the course of the program’s execution.

After its conversion, ENIAC’s programming system embodied these key features of the modern code paradigm as described in the 1945 First Draft. Its code and flow diagrams, which we describe in the third article in this series, have an unmistakable kinship with those described by von Neumann’s team in their publications on their IAS computer.

However, 1948 ENIAC only emulated (to use an anachronistic term) the von Neumann architecture and clearly did not acquire any of the minimalism of the EDVAC hardware paradigm. It was not a significant model for builders of later computers. We reported in our previous article that the significance of “EDVAC-type machines” in the 1940s was understood primarily in terms of the simplicity and flexibility of the new architecture that, combined with new memory technologies, promised a radical reduction in the number of expensive and unreliable tubes needed. Clearly, ENIAC remained just as bulky and full of tubes after its conversion, and thus it stayed off the mental radar of other computing groups. In contrast, the Manchester Baby’s demonstration that cathode-ray tube storage worked was treated as sensational news.⁵⁷

Comparative Analysis

When comparing the Manchester Baby, EDSAC, Pilot Ace, and 1948 ENIAC against von Neumann’s EDVAC design, it is only by elevating certain aspects and ignoring others that one can draw a clear line separating 1948

ENIAC from the other three machines (see Tables 1–3).

Although the Manchester Baby is universally recognized as a stored-program computer, the 1948 ENIAC came much closer to many aspects of the EDVAC vision. For example, von Neumann insisted on a large memory capacity, favoring 8,192 32-bit words. Yet the Manchester Baby had just 32 words of memory in which to store programs and data. It could run only programs of trivial scope. ENIAC’s writable memory was similarly tiny, but its read-only program memory was much closer to the size suggested in the report.

Once EDVAC-type machines were actually built, their users quickly discovered techniques to load program code into memory under computer control from external storage devices (considered as a separate organ, “R,” in von Neumann’s draft). This was crucial to many kinds of systems software and some automatic programming tools. Clearly, ENIAC could not do this, although EDVAC as described in the First Draft would also have faced some significant limitations because its instructions were flagged to protect them from being fully overwritten. As a matter of practice, both 1948 ENIAC and the Baby were usually programmed with switches. The Baby had no other input device, whereas 1948 ENIAC could execute programs directly from punched cards. The Baby’s parts were soon used to build a complete and useful computer, now known as the Manchester Mark I, which was fully operational by late 1949. We focus here on the Baby, however, because it is generally accepted as the first “stored program” computer to operate or, in some formulations, the first machine to run a stored program, and it hence provides a natural comparison point.

The question of utility has played a somewhat inconsistent role in the historical literature on early stored-program computers, perhaps because it has been used instrumentally to grant both Manchester and Cambridge the right to celebrate the first modern computer. EDSAC is generally called the first “useful,” “practical,” or “full-scale” stored-program computer and is thus credited with a more significant historical role than that of the Baby.⁷¹ Yet 1948 ENIAC was, for many kinds of problem, more useful, more practical, and larger in scale than either.

The instruction set of the 1948 ENIAC was comparable in scope to that of EDSAC, and of course, far more complete than that of the Manchester Baby. Most ENIAC instructions

Table 1. Part 1 comparison of ENIAC, EDVAC, and three other computers of the late 1940s with von Neumann's 1945 plan.

Computer	Operational (specs as of)	Programs loaded via	Programs usually executed from	Programs also executed from	Readable, addressable memory size*	Writable high-speed memory size*
1945 EDVAC	N/A	Mechanism not specified, storage was "organ R"	Mercury delay lines	N/A	8,192 32-bit words [262,000]	Same as readable
1945 ENIAC	Dec. 1945/ Jan. 1946	Rewired from "set up diagrams"	Distributed system of switches, plug boards and ad hoc busses	N/A	4,000 decimal digits (data only) [12,800]	200 decimal digits [640]
IBM SSEC	Jan. 1948	Run directly from card/tape	High-speed paper tape	Relay memory, electronic memory	Up to 100 19-digit words (only 8 electronic, others relay) ⁵⁸ [7,700 ⁵⁹]	Same as readable
1948 ENIAC	12 Apr. 1948	Turning switches	Banks of switches	Direct from punched card	4,000 decimal digits [12,800 ⁶⁰]	200 decimal digits (many double as special-purpose registers) [640]
Manchester Baby SSEM	21 June 1948	Flipping switches	Williams tubes	N/A	32 words [1,024]	Same as readable
EDSAC	6 May 1949	5-channel paper tape, 31 words of memory wired onto terminals for "initial orders"	Mercury delay lines	N/A	512 17-bit words ⁶¹ [8,704]	Same as readable

* Numbers in brackets indicates the approximately equivalent size in bits.

were stored in just two digits, giving it the potential to store a more complex program than EDSAC, which had to commit the entirety of one of its 512 17-bit words to store even the simplest instruction. Programming 1948 ENIAC involved a lot of shuffling data into and out of particular accumulators, but this was typical of the era and follows von Neumann's approach to registers in EDVAC. The compact two-digit instruction format of 1948 ENIAC included some unusually powerful instructions, including square root, a range of decimal shift options, and transfer of an entire punched card's worth of information to or from electronic memory.

Discussion of the computational legacies of early computers can easily veer into the

counterfactual. Fortunately, the ENIAC Monte Carlo calculations are exceptionally well documented, allowing us to focus on the abilities of the machine as actually used. The second version of the program consisted of approximately 840 instructions, largely filling two of ENIAC's function tables, and used most of the third for constant data. The code listing separates these into dozens of blocks, each numbered according to the corresponding operation on the flow diagram.⁷² Within this code, we have documented the use of conditional and unconditional jumps, a subroutine called from two points in the code, storage of return addresses, address modification via indirect addressing, the calculation of addresses for data reads to pull information

Table 2. Part 2 comparison of ENIAC, EDVAC, and three other computers of the late 1940s with von Neumann's 1945 plan.

Computer	Add time (micro-seconds)	Input/output	Conditional branch mechanism	Address modification mechanism
1945 EDVAC	N/A	Various options	Via instruction modification	Instruction modification
1945 ENIAC	200 ⁶²	Punched-card machines (one at 133 cps for output, one at 133 cps for input) ⁶³	Adapter and "dummy program" turns data output from comparison into control input	N/A
IBM SSEC	285 ⁶⁴	Many card readers and high-speed 80-track tape readers	Transfer control to device specified by 2-digit code, mapped via plug board	Instructions stored in electronic or relay memory could be modified programmatically
1948 ENIAC	1,200 ⁶⁵	Punched-card machines (one at 133 cps for output, one at 133 cps for input) ⁶⁶	N3D6 order loads 3-digit argument into accumulator 6, CT order jumps to that address if accumulator 15 is positive	Indirect addressing – the address to use for jumps and data reads is stored in a special register (a designated accumulator)
Manchester Baby SSEM	2,880 ⁶⁷	Results read from tube. Numbers entered via switches.	Skip next instruction if accumulator is negative.	Transfer instructions use memory indirect addressing, so modify the data word holding the address. ⁶⁸ For data operations via instruction modification
EDSAC	1,500	5 channel paper tape at 6.23 cps, ⁶⁹ Teleprinter	Conditional branch to location specified in instruction based on sign of accumulator or via instruction modification	Instruction modification

from one- and two-dimensional arrays stored in the function tables, and storage of program code and data within the same addressable memory space. We discuss these programs in detail in the final article in this series, "Gambling on ENIAC," describing the initial series of Monte Carlo calculations.⁷³ Thus, we feel comfortable asserting that ENIAC ran the first modern computer program.

What ENIAC still could not do was store variables in the same memory as code and constants. These had to be squeezed into the vacuum-tube memory spread around its 20 accumulators. In the Monte Carlo code, some accumulators were assigned to track values of the same physical quantities throughout the computation, rather like global variables, while others were reused for temporary storage of different variables at different

points in the calculation. In the conversion design as implemented, 131 decimal digits of accumulator memory were entirely safe from the threat of being overwritten and so could be used without any special precautions to store global data. Another 30 digits could be used with caution to store temporary data because their special roles were connected with operations such as square root that were executed infrequently.

A comparison with EDSAC gives a useful sense of how 1948 ENIAC measured up to the earliest universally recognized full-scale stored-program computers. The Monte Carlo code suggests that ENIAC's practical capabilities, at least for this task, were little affected by the restrictions imposed by its dual-memory system. It's unclear if EDSAC, as configured in 1949, could have accommodated

Table 3. Part 3 comparison of ENIAC, EDVAC, and three other computers of the late 1940s with von Neumann's 1945 plan.

Computer	Instruction format	Instruction word length	Instruction set	Maximum program length
1945 EDVAC	1 address	32 bits	8 operations, mostly load and store for different registers with one arithmetic instruction performing 1 of 10 operations on a stack-like configuration of three registers	8,192 instructions, less space needed for data
1945 ENIAC	N/A	N/A	N/A	N/A (multiple constraints distributed around machine)
IBM SSEC	4 address	19 decimal digits	2 instructions per word, 3 digits for operation code, 2 digits each for addresses, including next instruction source (have not located full instruction set)	Effectively unlimited (as a large program can span many paper tapes), each subroutine on a different tape, number of jumps limited by 2-digit address coding scheme
1948 ENIAC	0 and 1 address mix ⁷⁰	2 to 8 decimal digits	79 instructions, 40 of which are store or load variants and 20 are shifts	1,460 instructions (assumes average 2.6 digits per instruction), less space needed for constants
Manchester Baby SSEM	1 address	16 bit	Subtract, negate, transfer, conditional skip, conditional jump, unconditional jump, halt	32 instructions, less space needed for data
EDSAC	1 address	17 bit	Add, subtract, copy to multiply register, multiply (x2), transfer (x2), collate, shift (x2), conditional jump (x2), read tape, print, verify output, no operation, round, halt	512 instructions, less space needed for data (and initial orders)

this program at all. The 1948 Monte Carlo code was more than 800 instructions long, so an EDSAC conversion would have had to be streamlined before it could be loaded into its 512-word memory.⁷⁴ If program and constants could be compressed enough to fit at all, there would certainly be no more room for working data than on ENIAC.

Only ENIAC's relatively robust input and output capabilities made Monte Carlo feasible at all. A data-intensive simulation would soon have exhausted the memory of any early computer, leaving overall throughput bound by input-output performance. ENIAC could read and write punched cards using high-performance IBM units, and the Monte Carlo cards were sorted and processed between computer runs on conventional punched-card equipment. Like many other

early computers, EDSAC used paper tape, which was slower, cumbersome in large volumes, and impossible to sort. Hence, we doubt that these Monte Carlo calculations could have been usefully run on EDSAC, even if the code and data needed to run the program could have been shoehorned in.

Usability is another aspect of practicality. For a machine of its era, the 1948 ENIAC was, when working reliably, exceptionally easy to program and debug. ENIAC had always supported single-step debugging and a variable clock so that the machine could operate slowly enough for its operator to follow the progress of a computation. Neon lights in each accumulator displayed the entire contents of the electronic memory. Following the conversion to the modern code paradigm, the entire program was now visible on the

**ENIAC was the only
computer doing useful
work with programs coded
in the new style from April
1948 until early 1950.**

function tables and could be altered at any time by turning switches. This made it easy to set a break point or debug interactively.

Conclusions

Ulam was right to be excited. Thanks to the almost miraculous feat of user-driven innovation engineered by von Neumann’s collaborators, ENIAC acquired most, but not all, of the functional capabilities then associated with EDVAC-type machines. The Monte Carlo simulation run for Los Alamos in the spring of 1948 was written in the style of other early programs for modern computers, which we call the modern code paradigm, and was the first such program to be executed. ENIAC gained the flexibility and conceptual simplicity of programming associated with this paradigm, with program instructions and constant data held in the same read-only memory. This shows the remarkable malleability of this early computer and the profound role played by its users in reconstructing it.

Broadening our perspective from the architectural details of early computers and their relative positions on a timeline of seminal accomplishments to their uses further underscores the importance of 1948 ENIAC. ENIAC was the only computer doing useful work with programs coded in the new style from April 1948 until EDSAC began to regularly tackle problems from users in early 1950.⁷⁵ Not until SEAC began service in September 1950 was a more powerful computer of this kind available in the United States. We are confident that, as historians gravitate to issues of programming practice and computer use, the 1948 ENIAC conversion and its execution of the complex Monte Carlo stored program will increasingly be viewed as

important events in the early modern period of electronic computing.

Acknowledgments

This project was generously funded by Mrs. L.D. Rope’s Second Charitable Settlement. We thank archivists Susan Dayall at Hampshire College, Lynn Catanese at the Hagley Museum and Library, Susan Hoffman at the Charles Babbage Institute, Valerie-Ann Lutz and the other archival staff at the American Philosophical Society, and the staff of the Library of Congress Manuscripts Reading Room. Nate Wiewora, Alan Olley, and Peter Sachs Collopy provided us with copies of documents. Susan Abbey provided handwriting analysis services to clarify the authorship of numerous documents. William Aspray, Jeffrey Yost, Atsushi Akera, David Hemmendinger, Paul Ceruzzi, Simon Lavington, and Martin Campbell-Kelly kindly answered our questions on specific topics and shared their perspectives on computing in the 1940s. Thanks to all who discussed a late draft of this article at the Workshop on Early Programming Practices held in Milan in July 2013.

References and Notes

1. T.J. Bergin, ed., *50 Years of Army Computing: From ENIAC to MSRC*, US Army Research Laboratory, 2000.
2. J. Van der Spiegel et al., “The ENIAC - History, Operation and Reconstruction in VLSI,” *The First Computers: History and Architectures*, R. Rojas and U. Hashagen, eds., MIT Press, 2000, pp. 121–217; T. Zoppke and R. Rojas, “The Virtual Life of the ENIAC: Simulating the Operation of the First Electronic Computer,” *IEEE Annals of the History of Computing*, vol. 28, no. 2, 2006, pp. 18–25.
3. M. Bullynck and L. De Mol, “Setting-Up Early Computer Programs: D.H. Lehmer’s ENIAC Computation,” *Archive for Mathematical Logic*, vol. 49, no. 2, 2010, pp. 123–146; B.J. Shelburne, “The ENIAC’s 1949 Determination of π ,” *IEEE Annals of the History of Computing*, vol. 34, no. 3, 2012, pp. 44–54.
4. Erasure of women from earlier histories of ENIAC is discussed in J.S. Light, “When Computers Were Women,” *Technology and Culture*, vol. 40, no. 3, 1999, pp. 455–483. Their role was discussed in detail in W.B. Fritz, “The Women of ENIAC,” *IEEE Annals of the History of Computing*, vol. 18, no. 3, 1996, pp. 13–28, and presented

- to the public in the 2010 film *Top Secret Rosies: The Female "Computers" of WWII*.
5. M.S. Mahoney, "The Histories of Computing(s)," *Interdisciplinary Science Rev.*, vol. 30, no. 2, 2005, pp. 119–135.
 6. A. Aker, *Calculating a Natural World: Scientists, Engineers, and Computers during the Rise of U.S. Cold War Research*, MIT Press, 2007; S. McCartney, *ENIAC: The Triumphs and Tragedies of the World's First Computer*, Walker & Co, 1999.
 7. The most substantial general account of the ENIAC in use at BRL is W.B. Fritz, "ENIAC - A Problem Solver," *IEEE Annals of the History of Computing*, vol. 16, no. 1, 1994, pp. 25–45. Its role in meteorological calculations has been documented in P. Edwards, *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*, MIT Press, 2010, and K.C. Harper, *Weather by the Numbers: The Genesis of Modern Meteorology*, MIT Press, 2008. ENIAC's Monte Carlo work has been discussed in P. Galison, "Computer Simulation and the Trading Zone," *The Disunity of Science: Boundaries, Contexts, and Power*, P. Galison and D. J. Stump, eds., Stanford Univ. Press, 1996, pp. 118–157, and G. Dyson, *Turing's Cathedral: The Origins of the Digital Universe*, Pantheon Books, 2012.
 8. S. Ulam to J. von Neumann, letter, 12 May 1948, Papers of John von Neumann, Manuscripts Division, US Library of Congress (hereafter cited as JvN-LOC).
 9. This particular formulation is associated with Eric von Hippel, and the theme has been explored with particular respect to the history of computing in J. Yates, *Structuring the Information Age*, Johns Hopkins Univ. Press, 2005, and D. Caminer et al., eds., *User Driven Innovation: The World's First Business Computer*, McGraw-Hill Book Company, 1996.
 10. T. Haigh, M. Priestley, and C. Rope, "Reconsidering the Stored-Program Concept," *IEEE Annals of the History of Computing*, vol. 36, no. 1, 2014, pp. 4–17.
 11. This is taken from a page preserved among legal papers in series 10, box 3 of Herman H. Goldstine Papers, American Philosophical Soc. (hereafter cited as HHG-APS).
 12. The phrase comes from Metropolis himself, who wrote that after his successful runs in early 1948, "other Laboratory staff members made their pilgrimages to ENIAC to run Monte Carlo problems." N. Metropolis, "The Beginning of the Monte Carlo Method," *Los Alamos Science*, vol. 15, 1987, pp. 122–143.
 13. Von Neumann's work on computing is the central topic of W. Aspray, *John von Neumann and the Origins of Modern Computing*, MIT Press, 1990. His role has recently been explored in Dyson, *Turing's Cathedral*.
 14. J. von Neumann to S. Ulam, letter, 27 Mar. 1948. Stanislaw M. Ulam Papers, Am. Philosophical Soc., series 1, von Neumann, John #2.
 15. Fritz, "ENIAC – A Problem Solver," p. 31.
 16. J.P. Eckert et al., "Description of the ENIAC and Comments on the Electronic Digital Machine," AMP Report 171.2R, Applied Mathematics Panel, Nat'l Defense Research Committee, Moore School of Electrical Eng., 30 Nov. 1945, p. B-4.
 17. The report on "Planning and Coding Problems for an Electronic Computing Instrument" was issued in installments during 1947 and 1948 and is reprinted in J. von Neumann, *Papers of John von Neumann on Computing and Computing Theory*, W. Aspray and A. Burks, eds., MIT Press, 1987.
 18. H.H. Goldstine, *The Computer from Pascal to von Neumann*, Princeton Univ. Press, 1972, p. 270. This date may be suspect because one of his letters dated 28 July notes, "Adele just received a duly signed and executed contract from Los Alamos. So she is now officially in business." H. Goldstine to J. von Neumann, 28 July 1947, JvN-LOC, box 4, folder 1.
 19. Klara von Neumann shifted from salary to hourly compensation at the end of August 1947, but it is unclear exactly when she was first hired. Kelly to Richtmyer, 28 Aug. 1947, JvN-LOC, box 19, folder 7.
 20. A.K. Goldstine, *A Report on the ENIAC Part I: Technical Description of the ENIAC*, vol. I, Moore School, Univ. of Pennsylvania, 1946, VII.21-VII.24; J.P. Eckert and J.W. Mauchly, "Electronic Numerical Integrator and Computer," US Patent 3120606, 4 Feb. 1964.
 21. R.F. Clippinger, "Oral History Interview with Richard R Mertz," 1 Dec. 1970, Computer Oral Histories Collection. His claim has been repeated by others, including J.J. Bartik, *Pioneer Programmer: Jean Jennings Bartik and the Computer that Changed the World*, Truman State Univ. Press, 2013.
 22. R.F. Clippinger, "A Logical Coding Scheme Applied to the ENIAC," BRL report no. 673, Aberdeen Proving Ground, 1948.
 23. Clippinger Testimony, Honeywell, Inc. vs. Sperry Rand Corp., et al. 22 Sept. 1971, pp. 8952–8968 in ENIAC Patent Trial Collection, 1864 – 1973, Univ. of Pennsylvania Univ. Archives and Records Center.
 24. J.P. Eckert, "The ENIAC," *A History of Computing in the Twentieth Century*, N. Metropolis, J. Howlett, and G.-C. Rota, eds., Academic Press, 1980, 525–539, p. 529.

25. The other members were Arthur Gehring, Ed Schlain, Kathe Jacobi, and Sally Spear. Bartik, *Pioneer Programmer*, pp. 115–116.
26. J. Bartik, interview with G. Hendrie, 1 July 2008, Computer History Museum; http://archive.computerhistory.org/resources/text/Oral_History/Bartik_Jean/102658322.05.01.acc.pdf.
27. J. von Neumann to R.H. Kent (BRL), 13 June 1947, JvN-LOC, box 5, folder 13.
28. Goldstine’s travel records confirm visits to Aberdeen on 29 Aug. 1947 and to the Moore School to visit Bartik’s group on 7 Oct. and 17 Oct. A. Goldstine, “Travel Expense Bill,” 17 Dec. 1947, HHG-APS, series 7, box 1.
29. “Control Code for ENIAC”, 10 July 1947, HHG-APS, series 10, box 3.
30. The most detailed discussion in recent decades of ENIAC instruction capabilities post-conversion is in H. Neukom, “The Second Life of ENIAC,” *IEEE Annals of the History of Computing*, vol. 28, no. 2, 2006, pp. 4–16, and particularly its “Web extras” online technical supplement.
31. Ballistic Research Laboratories, “Technical Note 141: Description and Use of the ENIAC Converter Code,” Aberdeen Proving Ground, 1949, p. 9.
32. The design for the IAS computer was described in a highly influential technical report, A.W. Burks, H.H. Goldstine, and J. von Neumann, *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*, Inst. for Advanced Studies, 1946.
33. H.H. Goldstine and J. von Neumann, *Planning and Coding Problems for an Electronic Computing Instrument*, part II, vol. 1, Inst. for Advanced Studies, 1947, section 7.
34. Bartik later described this work in Bartik, *Pioneer Programmer*, pp. 113–120. She writes that the contract began in March 1947. Her memoir can be read as claiming that the group was originally created to assist Clippinger with the conversion work, but given the timeline we find it more likely that the group was established to assist Clippinger with his airflow simulation problems and that the conversion project was launched just in time to occupy Bartik’s energies for the first part of the contract. A claim on the book’s flap that “In 1946, Bartik headed up the team that modified the ENIAC into the first stored-program computer” is faithful neither to reality nor to the book’s actual text.
35. This stepper was one of a number of new pieces of ENIAC hardware commissioned by BRL in 1946. Design work was well advanced by June 1947, as evidenced by blueprints PX-4-122 (10 June), PX-4-212 (2 July), and PX-4-125 (16 July) 1947. These are all from box 19, Sperry Rand Corporation, Univac Division records (Accession 1825.I), Hagley Museum and Library. This collection is cited hereafter as HL-UV.
36. This is the version included in R.F. Clippinger, *A Logical Coding Scheme Applied to the ENIAC*, BRL report no. 673, Aberdeen Proving Ground, 1948. The document has a cover publication date of 29 Sept. 1948 and an interior date of 30 July 1948 on the abstract page. However, pages 20–23 are titled “Orders for 60 Word Vocabulary – November 13, 1947.”
37. Anonymous, “60 Order Code, Nov 21 – 1947”, Herman H. Goldstine Collection, 1941–1971, archives, Hampshire College, box 1, folder 5. This collection is hereafter cited as HHG-HCA.
38. Anonymous, “Testing ENIAC – 60 Order Code,” HHG-HCA, box 1, folder 8.
39. BRL’s decision to install the memory unit must have been taken by 18 Feb. because this was the date it communicated the news to the Moore School. L. Simon to I. Travis, letter, 18 Feb. 1947, box 51, Moore School of Electrical Eng. Office of the Director Records, 1931–1948, Univ. of Pennsylvania Archives. This collection is cited hereafter as UP-MSOD. After many delays it was finally installed on 27 May 1949, but it proved unusable. “ENIAC Operations Log,” box 10, HL-UV.
40. R.F. Clippinger, “Adaption of ENIAC to von Neumann’s Coding Technique (Summary of Paper Delivered at the Meeting of the Association for Computing Machinery, Aberdeen, MD, Dec 11-12 1947) - Plaintiff’s Trial Exhibit Number 6341,” 1948, ENIAC Trial Exhibits Master Collection (CBI 145), Charles Babbage Inst., Univ. of Minnesota.
41. W. Lisner, “‘Brain’ Speeded Up, For War Problems! Electronic Computer Will Aid in Clearing Large Backlog in Weapon Research,” *New York Times*, 12 Dec. 1947.
42. J. von Neumann to L. Simon, letter, 5 Feb. 1948, HHG-APS, series 1, box 3.
43. N. Metropolis and J. Worlton, “A Trilogy on Errors in the History of Computing,” *Annals of the History of Computing*, vol. 2, no. 1, 1980, pp. 49–59. On page 54 Metropolis recalled having originated the idea of using the “converter” for decoding when, “[o]n a preliminary visit to the Aberdeen Proving Ground in Maryland,” he “noticed a complete many-to-one decoder network nearing completion; it was intended to increase the capability of executing iterative loops in a program.” “We dated this trip to 20 Feb. using the ENIAC Operations Log from HL-UV. BRL staff continued to work on the 60 order code into March, so the decision by

Metropolis and von Neumann to use the converter to enable the full range of two-digit codes unquestionably diverged from BRL's established plans. However the idea of using the full range of codes predated their 20 Feb. visit as occasional references appear to projected development of a 99 order code apparently intended for use with the register memory and its unresolved "problems" as early as 19 Jan. 1948. This would also have required use of the converter for decoding. Thus, although Metropolis was responsible for reworking plans for the interim reconfiguration to use the converter, this probably rediscovered and implemented a technique already envisioned for eventual use once the register arrived (then expected to occur around May 1948).

44. We have not yet been able to confirm the original intended function of the converter. BRL ordered changes to ENIAC installation plans to make room for it, and the previously discussed "register" delay line memory unit, on 18 Feb. 1947. L. Simon to H. Pender, letter, 18 Feb. 1947, UP-MSOD. This and the register were referred to collectively as the "Automatic Program Selector": in "MS 111 Moving ENIAC Progress Report 1 March 1947," box 51, UP-MSOD. We therefore suspect that it was intended to realize an extension of the subroutine sequencing technique discussed in the 1946 ENIAC documentation. Eckert and Mauchly hoped to win the contract to provide these items for their fledgling Electronic Control Corporation and may have been responsible for selling the idea to BRL staff. Untitled Transcript of Negotiations Between Travis and Dederick, box 51, UP-MSOD. Blueprints show that the converter had been fully designed by 19 Aug. 1947. C. Shearman, "PX-16-304: Converter Front Panel, 3-7-47" and "PX-16-301: Converter Block Diagram, 8-19-47" both box 2, HHG-HCA.
45. Previous accounts have suggested that ENIAC was operated using the 60 order code for some time prior to the addition of the converter and shift to the full 100 order code. For example, Neukom, "The Second Life of ENIAC," and Fritz, "ENIAC – A Problem Solver."
46. S. Ulam to J. von Neumann, letter, 12 May 1948, JvN-LOC.
47. ENIAC Operations Log, 17 and 18 May 1948.
48. "Description of Orders for Coding ENIAC," 6 July 1948, HHG-HCA, box 1.
49. Accounts differ regarding whether the ENIAC was ever temporarily "reconverted" to run the ballistics calculations already programmed using its native mode. Clippinger recalled that this did take place in an oral history with R.R. Mertz, 1 Dec. 1970 held in the Computer Oral Histories collection of the National Museum of American History. However, we found no evidence in the log book that this had taken place during the period covered (through August 1949), and an order from B. Dimsdale to carry this out in February 1949 was ignored by ENIAC staff following protests from John von Neumann that the ENIAC's unique capabilities were crucial to the AEC and would be jeopardized by a "double-changeover [that] can and probably will consume a great deal more time than one may optimistically estimate," J. von Neumann to Kent, 16 Mar. 1949, JvN-LOC, box 12, folder 3.
50. A total of 87 problems are listed: W.B. Fritz, "BRL Memorandum Report No. 617: A Survey of ENIAC Operations and Problems: 1946–52," Ballistics Research Laboratory, 1952. An earlier list, ordered chronologically, identifies the Los Alamos Monte Carlo as the 13th problem tackled by ENIAC. Anonymous, "List of Problems That the Eniac Has Been Used to Solve," circa 1949, exhibit 44 in "Sperry Rand v. Bell Telephone Laboratories, Defendant's Goldstine Exhibit," HHG-APS, series 10, box 3. Thus, it appears that 75 problems were tackled using code written in the new paradigm over the first four years of operation.
51. Metropolis and Worlton, "A Trilogy on Errors in the History of Computing." This was originally presented at a conference in 1972. At that point, Metropolis may not have been aware of the Manchester Baby.
52. Goldstine, *The Computer from Pascal to von Neumann*, p. 233.
53. Neukom, "The Second Life of ENIAC." Two accounts based on different primary sources support Metropolis's claim of a spring 1948 date for the first Monte Carlo calculations and hence for the initial conversion over Goldstine's later date: Aspray, *John von Neumann and the Origins of Modern Computing*, 238–239 (although p. 112 gives an incorrect 1947 date) and A. Fitzpatrick, *Igniting the Light Elements: The Los Alamos Thermonuclear Weapon Project, 1942–1952* (LA-13577-T), Los Alamos Nat'l Laboratory, 1999, p. 268. However only C. Rope, "ENIAC as a Stored-Program Computer: A New Look at the Old Records," *IEEE Annals of the History of Computing*, vol. 29, no. 4, 2007, pp. 82–87, has drawn explicit attention to the contradiction between this evidence and Goldstine's date.
54. Goldstine, *The Computer from Pascal to von Neumann*, p. 233.
55. Aspray, *John von Neumann and the Origins of Modern Computing*, pp. 238–239.
56. One of us addressed the role of Turing in the invention of the "stored program concept" in

- T. Haigh, “Actually, Turing Did Not Invent the Computer,” *Comm. ACM*, vol. 61, no. 1, 2014, pp. 36–41.
57. The influence of the Williams Tube on the work of von Neumann’s group at the Institute for Advanced Studies is related in Dyson, *Turing’s Cathedral*, pp. 142–148.
 58. SSEC had 150 words of relay memory and 8 words of electronic memory, but each location from which an instruction was to be read had to be given its own two-digit code. These codes were mapped to actual memory locations via a plug board. In practice, only instructions requiring modification were read from relay or electronic memory. Branches were accomplished via the transfer of control to a different paper-tape drive, from which instructions could be read sequentially without the need for multiple plug board entries. Each subroutine or inner loop was stored on a different drive.
 59. SSEC stored 19 decimal digit signed words in binary coded decimal format as 76 number bits plus sign and parity, so we are treating each word as 77 data bits here.
 60. Conversion assumes that one 20-digit decimal word is equivalent to one 64-bit word. Use of binary-coded decimal would require 4 bits per digit, or 80 bits per 20-digit word.
 61. The original memory size of EDSAC is sometimes given as 256 36-bit words, reflecting its ability to handle 35-bit data words as well as 17-bit instruction words. Because of timing issues with its delay line memory, the last bit in either size of word was always wasted.
 62. 1945 ENIAC is hard to compare here because several additions could be performed simultaneously.
 63. According to Goldstine, *A Report on the ENIAC Part I: Technical Description of the ENIAC*, vol. I, VII-1: “The rate is 160 cards per minute when the reader reads continuously without stopping and may be either 120 or 160 cards per minute when the reader stops between readings.” For output, Goldstine notes (IX-1) that “Cards may be punched at the rate of approximately 100 per minute.”
 64. However, SSEC would usually spend a much longer time waiting for its next instruction to load from high-speed paper tape, an issue only partly alleviated by a complex system of loading instructions two at a time. Even reads from relay memory could not take place at electronic speeds, because relays had to move to access the location needed.
 65. Confusingly, after conversion, ENIAC needed six “add times,” each 200 microseconds, to execute one addition. Subtraction took 8, multiplication 20, and shifts 9. We are assuming a 100 KHz operation here. Later improvements reduced these times.
 66. Instructions to punch (Pr) or read (Rd) a card each took 3,000 add times—600,000 microseconds, or 0.6 seconds. Each card held 80 characters, and hence throughput of 133 characters per second.
 67. Adding required two operations because it was performed via subtraction and negation.
 68. A transfer instruction on the Baby specified the address from which to fetch the address to jump to, either without further alteration or with the addition of the old contents of the program counter.
 69. As of 1949, “Input was on 5-track teleprinter paper tape via an electromechanical tape reader running at 6 2/3 characters per second.... Output was delivered via a teleprinter at 6 2/3 char/sec.,” www.cl.cam.ac.uk/conference/EDSAC99/statistics.html.
 70. As discussed previously, ENIAC had different operation codes to store data in and retrieve from each accumulator. Reads from the function table used an address previously stored in the function table, as did jump instructions. The instructions storing those numbers were the only ones that literally included an address, although several others were followed by arguments.
 71. Descriptions, respectively, from P.E. Ceruzzi, *Computing: A Concise History*, MIT Press, 2012, p. 50; M. Campbell-Kelly and W. Aspray, *Computer: A History of the Information Machine*, Basic Books, 1996, p. 104; and M. Campbell-Kelly and W. Aspray, *Computer: A History of the Information Machine*, Basic Books, 1996, photographic inset.
 72. “Card Diagram, FLOW DIAGRAM, Coding, Function Table III Values, Monte Carlo Second Run,” 1948, JvN-LOC. The cover page is alas misleading, as only the coding remains.
 73. T. Haigh, M. Priestley, and C. Rope, “Los Alamos Bets on ENIAC: Nuclear Monte Carlo Simulations, 1947–1948,” to appear in *IEEE Annals of the History of Computing*, vol. 36, no. 3, 2014.
 74. After comparing the instruction sets of the two machines and looking at the program code we see no reason to believe that an EDSAC version of the code would have needed significantly fewer instructions.
 75. One “key milestone in the life of the EDSAC” was “Early 1950: regular use begins.” M. Campbell-Kelly, “Programming the EDSAC: Early Programming Activity at the University of Cambridge,” *Annals of the History of Computing*, vol. 2, no. 1, 1980, pp. 7–36, quote from p. 7. Clarifying this, Campbell-Kelly wrote that, “By

early 1950, a programming system had been set up and the EDSAC began to attract a number of regular users," p. 8.



Thomas Haigh is an associate professor of information studies at the University of Wisconsin-Milwaukee. His research interests include the history of computing, especially from the viewpoints of labor history, history of technology, and business history. Haigh has a PhD in the history and sociology of science from the University of Pennsylvania. See more at www.tom-andmaria.com/tom. Contact him at thaigh@computer.org.



Mark Priestley is an independent researcher into the history and philosophy of computing, with a special interest in the development of programming. He started his career as a programmer and was for many years a lecturer in software engineering at the University of

Westminster before turning to the history of computing. Priestley has a PhD in science and technology studies from University College London. His most recent book, *A Science of Operations: Machines, Logic, and the Invention of Programming* (Springer, 2011), explores the coevolution of programming methods and machine architecture. More information is available at www.markpriestley.net. Contact him at m.priestley@gmail.com.



Crispin Rope has been interested in ENIAC since reading Douglas Hartree's pamphlet on the machine from 1947 and has pursued an a vocational interest in its history for more than a decade. His earlier work on this topic has been published in *IEEE Annals of the History of Computing* and *Resurrection: The Bulletin of the Computer Conservation Society*. Contact him at westerfield@btconnect.com.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

Computing History Anecdotes Wanted

The Anecdotes column of *IEEE Annals of the History of Computing* accepts stories of 3,000 words or less based on personal experience, observation, or study of a slice of computing history. If you have a story to tell from computing history, please email your idea (or draft) to anecdotes@computer.org

Visit www.computer.org/comphistory/anecdotes/ to see a list of titles that exemplify the variety of possible anecdote topics.



www.computer.org/comphistory/anecdotes/