The Undead of Information

Computers have conflated memory with storage, the ephemeral with the enduring. Rather than storing memories, we now put things "into memory," both consciously and unconsciously. "Memory"—computer memory—has become surprisingly permanent. As Matthew Kirschenbaum has argued, our digital traces remain far longer than we suppose.[1] Hard drives fail, but can still be "read" by forensic experts (optically, if not mechanically); our ephemeral documents and other "ambient data" are written elsewhere—that is "saved"—constantly. Again, to read information is to write it elsewhere. At the same time, however, the enduring is also the ephemeral. Not only because even if data storage devices can be read forensically after they fail they still eventually fail, but also because—and more crucially—what is not constantly upgraded or "migrated" or both becomes unreadable. As well, our interactions with computers cannot be reduced to the traces we leave behind. The experiences of using—the exact paths of execution—are ephemeral. Information is "undead": neither alive nor dead, neither quite present nor absent.

Memory and storage are different. Memory stems from the same Sanskrit root for *martyr* and is related to the ancient Greek term for baneful, fastidious. Memory contains within it the act of repetition: it is an act of commemoration—a process of recollecting or remembering. In contrast, a store, according to the OED, stems from the Old French term *estorer* meaning "to build, establish, furnish." A store—like an archive—is both what is stored and its location. Stores look toward a future: we put something in storage in order to use it again; we buy things in stores in order to use them. By bringing memory and storage together, we bring together the past and the future; we also bring together the machinic and the biological into what we might call the archive.

Sigmund Freud famously modeled the human memory system, which he posited as fundamentally unconscious, on a toy called the *Mystic Writing Pad*. Describing the device, he wrote:

> The surface of the Mystic [Writing] Pad is clear of writing and once more capable of receiving impressions. But it is easy to discover that the permanent trace of what was written is retained upon the wax slab itself and is legible in suitable lights. Thus the Pad provides not only a

> receptive surface that can be used over and over again, like a slate, but also permanent traces of what has been written like an ordinary paper pad . . . this is precisely the way in which, according to the hypothesis which I mentioned just now, our mental apparatus performs its perceptual function. The layer which receives the stimuli—the system *Pcpt.-Cs*. [Perception-Consciousness]—forms not permanent traces; the foundations of memory come about in other, adjoining, systems.[2]

According to Derrida, Freud, through this formulation posits a "prosthesis of the outside," which makes psychoanalysis a theory of the archive as well as of memory. It makes possible the "idea of an archive properly speaking, of a hypomnesic or technical archive, of a substrate or the subjectile (material or virtual) which, in what is already a psychic *spacing*, cannot be reduced to memory."[3] Memory in psychoanalysis is not first "live" and is not outside representation. Contemplating the importance of technology to this theory, Derrida asks, "Is the psychic apparatus *better represented* or is it *affected differently* by all the technical mechanisms for archivization and for reproduction . . . (microcomputing, electronization, computerization, etc.)?"[4] Intriguingly, the Mystic Writing Pad—or more properly its modern version, the Etch A Sketch®—returns as the model for the hard drive in a textbook on computer forensics. To explain the "unerasability" of hard drives, Warren G. Kruse II and Jay G. Heiser compare them to Etch A Sketches:

> When data is written onto magnetic media, a faint image of what was previously on the drive remains. A hard drive is like the child's drawing toy, the Etch A Sketch. Well, hard drives don't leak silver powder, but we are referring to the faint traces left after you erase an Etch A Sketch. The Etch a Sketch is erased by turning it over and shaking it, allowing the silver powder to coat the inside of the clear plastic window, preparing it for more drawings. But if you've used this popular toy, you'll remember that the faint traces of the previous drawing are always left behind. . . . Magnetic media—including hard drives—are similar in that every write leaves faint traces behind it, even when media have been overwritten numerous times.[5]

Data on a hard drive, Kruse and Heiser emphasize, leave a permanent trace, even as the drive makes room for new "impressions." This description of the hard drive, written by information security experts, eerily repeats Freud's description of the unconscious. It also highlights the work that "memory" (in contrast to archiving) entails—to be retrieved, these traces must be submitted to a rigorous process of reading.

How are we to understand archives as linking the machinic to the human to the written? As linking the ephemeral to the lasting? The alive to the dead? Two things to consider:

1. *The RNA world* As mentioned previously, scientists are considering RNA more and more as primary. What is called the RNA world thesis argues that RNA is the "origin" of life, since RNA can act as both genes and enzymes and because DNA replication depends on "an enormous amount of proteins" (thus making DNA as origin unlikely).[6] Through retroviruses, RNA also rewrites DNA. This thesis fascinatingly questions the

conflation of legislation with execution that grounds code as logos. RNA does not simply code for proteins; DNA is no simple source.

2. *Cybernetics as memory* Jacques Derrida, in *Of Grammatology*, linked together writing and cybernetics: "The entire field covered by the cybernetic *program* will be the field of writing. If the theory of cybernetics is by itself to oust all metaphysical concepts—including the concepts of soul, of life, of value, of choice, of memory—which until recently served to separate the machine from man, it must conserve the notion of writing, trace, grammè [written mark], or grapheme, until its own historico-metaphysical character is also exposed."[7] Cybernetics, however, did not only have to conserve the notion of writing, but also that of memory. Memory links together the man and the machine. Memory also bridges across the machinic and human unknowns.

Moreover, to understand information as undead, we need to understand its relation to that other undead thing—the commodity. If a commodity is, as Marx famously argued, a "sensible supersensible thing," information would seem to be its complement: a supersensible sensible thing.[8] The literature, of course, on the relationship between information and the commodity is dense: from procapitalist celebrations of information as the new commodity to neo-Marxist ruminations on the impact of information on labor practices. Rather than rehearse these arguments, I want to emphasize that this parallel between information (as a general, rather than technical term) and commodities intersects with the emergence of source code as information outlined in chapter 1. That is, if information is a commodity, it is not simply due to historical circumstances or to structural changes; it is also because commodities, like information, depend on a ghostly abstraction.

Thomas Keenan, in "The Point Is to (Ex)Change It: Reading *Capital* Rhetorically," unpacks Marx's use of ghostly rhetoric to explain capital, in particular the capitalist exchange. Abstraction, Marx argues, transforms material things and their embedded-use values, into things that can be exchanged: commodities. This transformation fundamentally changes the "atomic" structure of things: "as exchange-values, [things] can be only different qualities, and thus not contain an atom . . . of use-value." Keenan asks: What, after this abstraction, is left? If exchange value eviscerates use—if it must eviscerate use to work—what makes possible exchange? What remains, Keenan contends, is a "ghost, *gespenstige Gegenständlichkeit*, spectral, haunting, surviving objectivity. 'There is nothing of them left over but this very same . . . ghostly objectivity, a mere jelly . . . of undifferentiated human labor.'" "This very phantom," Keenan goes on to insist, "makes possible the relation between (or within) things or uses, grants the common axis of similarity hitherto unavailable, precisely because it is a ghost and no longer a thing or a labor."[9] That ghostly jelly, Keenan argues, is humanity—the common humanity that survives in the things exchanged and, like language, makes exchange possible.

4 Always Already There, or Software as Memory

Software—as instructions and information (the difference between the two being erased by and in memory)—not only embodies the always already there, it also grounds it. It enables a logic of "permanence" that conflates memory with storage, the ephemeral with the enduring. Through a process of constant regeneration, of constant "reading," it creates an enduring ephemeral that promises to last forever, even as it marches toward obsolescence/stasis. The paradox: what does not change does not endure, yet change—progress (endless upgrades)—ensures that what endures will fade. Another paradox: digital media's memory operates by annihilating memory.

Remarkably, digital media has been heralded as "saving" analog media from destruction and obscurity. Many users, blind to the limitations of electromagnetic materials, assume that one can actually "store" things in memory. They assume that data saved on their DVDs, hard drives, and jump drives will always be there, that disk failure and the loss of memory it threatens are accidents instead of eventualities. Digitization surprisingly emerged as a preservation method in the 1990s by becoming a major form of "reformatting," a procedure designed to save intellectual content threatened by decaying materials—such as acidic wood-pulp paper and silver-nitrate film—by reproducing it.[1] Indeed, the National Endowment for the Humanities' 1988 "Brittle Books Program," which microfilmed millions of books in peril of "slow burn," viewed digitization as the preferred preservation method, even given a computer file's five-year shelf life. This celebration of the digital as archives' salvation stems in part from how digital files address another key archival issue: access. From the Library of Congress's early attempt to digitize its collections, the American Memory Pilot program (1990–1994), to Google's plan to digitize over ten million unique titles through its Book Search Program (announced in 2004), digitization has been trumpeted as a way for libraries finally to fulfill their mission: to accumulate and provide access to human knowledge. Digital archives are allegedly H. G. Wells's "World Brain" and André Malraux's museum without walls, among other dreams, come true.

At the same time, however, computer archives have been targeted as *the source* of archival decay and destruction, their liquidity threatening both the possibility and the authenticity of cultural memory. Digital media disrupt the archive because they themselves are difficult to archive or have not been properly archived or both. The 1999 Modern Languages Association (MLA) report, "Preserving Research Collections: A Collaboration between Librarians and Scholars," summarizes the dual challenges of the hard and the soft: "Imagine a historian opening a late nineteenth-century text and helplessly watching as the title page breaks in her hand. Imagine another scholar, ten years from now, inserting a disk containing an important document into the computer and reading only a "fatal error" message on his screen. These two examples illustrate the Janus-like preservation challenge faced by research libraries today: fragility of the print past and the volatility of the future."[2] The material limits of materials not only cause the future to be volatile, but also, again, so do the ever-updating, ever-proliferating, and increasingly incompatible soft and hard technologies—the challenges to the historical preservation of software outlined in the introduction to this book. Moreover, digital imaging potentially destabilizes authenticity. If libraries and archives, as Abby Smith has argued, "serve not only to safeguard that information [which has long-term value], but also to provide evidence of one type or another of the work's provenance, which goes to establishing the authenticity of that work," this function is seriously undermined by electronic images and documents, which are easily changed or falsified.[3] The sheer plethora of digital files also calls into question the importance of the libraries' and archives' traditional gatekeeping function. This is most clear in the Internet Wayback Machine (IWM)'s approach to selection: this site creates a "library of the Internet" by backing up all accessible sites. If libraries and archives traditionally distinguished between materials of enduring value and "other bits of recorded information, like laundry lists and tax returns," which were allowed to vanish, the IWM has solved the extremely time-consuming task of selecting the enduring from the ephemeral by saving everything. (Although it originally tried to save only "significant" material, it soon became an automatic archive of everything.) In addition to all these difficulties, attempts to digitize content have been frustrated by copyright issues, with rights holders demanding compensation or refusing permission. Digital copies—allegedly defined by their immateriality—are, as the introduction has emphasized, more closely regulated than their material counterparts, especially since their use can be controlled by private contracts rather than by copyright or patents.

As this discussion makes clear, digital media's promise is also its threat; the two cannot be neatly divided into the good and the bad. Digital media, if it "saves" anything, does so by transforming storage into memory, by making what decays slowly decay more quickly, by proliferating what it reads. By animating the inanimate—crossing the boundary between the live and the dead—digital media poses new challenges and opportunities for "the archive."

Taking up the intertwining of the biological and the technological addressed previously, this chapter investigates how something as admittedly "soft" (and vapory) as software hardened into something that allegedly guarantees heredity, and permanence. Looking in particular at von Neumann's early formulation of stored-memory computer architecture, chapter 4 argues that memory became conflated with storage through analogies to analogies: through analogies to cybernetic neurons, to genetic programs, to what would become "analog" media itself. Through these analogies (and their erasure), the new and the different have been reduced to the familiar. I uncover these differences and analogies not to attribute blame, but rather to reveal the dreams and hopes driving these misreadings: the desire to expunge volatility, obliterate ephemerality, and neutralize time itself, so that our computers can become synonymous with archives.[4] These desires are key to stabilizing hardware so that it can contain, regenerate, and thus reproduce what it "stores." Further, they are central to the twin emergence of neoliberalism and computer programs as strategic games.

These analogies also ground one of the fundamental axioms of digital media, namely that the digital reduces the analog—the real world—to 1s and 0s. By doing so the digital allegedly releases and circulates information that before clung stubbornly to material substances, effectively erasing the importance of context and embodiment. The fact that this has become an axiom should make us pause, especially since the evidence against it is substantial: the digital has proliferated, not erased, media types; what has become the analog is not the opposite, but rather the "ground" of the digital; and last, information is not naturally or inherently binary. Rather than making everything universally equivalent, the digital has exploded differences among media formats. Proprietary and nonproprietary electronic file formats such as jpeg, gif, mp3, QuickTime, doc, txt, rtf, and so on, not only distinguish between image, sound, and text, but also introduce ever more numerous differences among them. This explosion is not accidental to the digital, but rather, as I argue later, central to it. Also, the term *analog*, based on the word *analogy*, does not simply refer to what is real. After the emergence of electronic, arithmetically based computers, the term *analog* was adopted to describe computers that solved problems using similar physical models, rather than numerical methods. And finally, information is not simply digital, for information stems from the transmission of continuous electronic signals. The information traveling through computers is not 1s and 0s; beneath binary digits and logic lies a messy, noisy world of signals and interference. Information—if it exists—is always embodied, whether in a machine or an animal. To make information appear disembodied requires a lot of work, work that is glossed over if we just accept the digital as operating through 1s and 0s.

Revising the working thesis of chapters 2 and 3—software as axiomatic—chapter 4 contends that the digital is axiomatic. The digital emerges as a clean, precise logic through an analogy to an analogy, which posits the analog as real/continuous. Looking at the differences between analog and digital computers, this chapter reveals how discrete logical devices work by restricting possibilities and possible decodings.

It also examines how the development of these devices drives the need for "memory," a regenerating and degenerating archive that paradoxically, as Geoffrey C. Bowker notes, annihilates memory by substituting generalized patterns for particular memories.[5] This does not simply erase human agency, however, but rather fosters new dreams of human intervention, action, and incantation. It does not absolve us of responsibility, but instead calls on us to respond constantly, to save actively, if we are to save at all.

Biological Abstractions

John von Neumann's mythic, controversial, and incomplete 1945 "First Draft of a Report on the EDVAC" introduced the concept of stored program computing and memory to the U.S. military and the academic "public." This report is remarkably abstract: rather than describing actually existing components, such as vacuum tubes and mercury delay lines, it offers "hypothetical elements." According to von Neumann, it does so because, although dealing with real elements such as vacuum tubes would be ideal, such specificity would derail the process by introducing specific radio engineering questions at too early a stage. Thinking concretely in terms of types and sizes of vacuum tubes and other circuit elements "would produce an involved and opaque situation in which the preliminary orientation which we are now attempting would be hardly possible." To avoid this, von Neumann bases his consideration "on a hypothetical element, which functions essentially like a vacuum tube—e.g., like a triode with an appropriate associated RLC-circuit—but which can be discussed as an isolated entity, without going into detailed radio frequency electromagnetic considerations."[6] The vagaries of the machinery (vacuum tubes etc.), which are not necessarily digital but can be made to act digitally, threaten the clean schematic logic needed to design this clean, logical machine. Von Neumann describes this deferral as "only temporary."[7] However, J. Presper Eckert and John Mauchly, the original patent holders of stored program computing, would allege that von Neumann did not touch on the "true electromagnetic nature" of the devices because it was outside his purview: von Neumann, they contended, merely translated their concrete ideas into formal logic.[8] In fact, rather than a temporary omission, abstractness was von Neumann's modus operandi, central to the "axiomatic" (blackboxing) method of his general theory of natural and artificial automata and consonant with his game theory work.

This fateful abstraction, this erasure of the vicissitudes of electricity and magnetism, surprisingly depends on an analogy to the human nervous system. As cited earlier, von Neumann specifies the major components of the EDVAC as corresponding to different neurons: "The three specific parts CA [central arithmetic], CC [central control] (together C) and M [memory] correspond to the associative neurons in the human nervous system. It remains to discuss the equivalents of the *sensory* or *afferent* and the *motor* or *efferent* neurons. These are the *input* and the *output* organs of the device."[9] These neurons, however, are not simply borrowed from the human nervous system. They are the controversial, hypothetical neurons postulated by Warren McCulloch and Walter Pitts in their "A Logical Calculus of Ideas Immanent in Nervous Activity," a text McCulloch claims von Neumann saved from obscurity.[10] (Von Neumann would later describe these neurons as "extremely amputated, simplified, idealized.")[11] In accordance with McCulloch and Pitts, von Neumann expunges the messy materiality of these "neurons":

> Following W. S. McCulloch and W. Pitts . . . we ignore the more complicated aspects of neuron functioning: thresholds, temporal summation, relative inhibition, changes of the threshold by after-effects of stimulation beyond the synaptic delay, etc. It is, however, convenient to consider occasionally neurons with fixed thresholds 2 and 3, that is, neurons which can be excited only by (simultaneous) stimuli on 2 or 3 excitatory synapses (and none on an inhibitory synapse). . . . It is easily seen that these simplified neuron functions can be imitated by telegraph relays or by vacuum tubes. Although the nervous system is presumably asynchronous (for the synaptic delays), precise synaptic delays can be obtained by using synchronous setups.[12]

This analogy thus depends on and enables a reduction of both technological and biological components to blackboxes. In this simplified analogy, the effects of time are ignored to the extent that the synchronous can substitute for the asynchronous and interactions or "after effects" are erased.

So: to what extent are these abstractions and analogies necessary? What did and do they make possible? Clearly, this blackboxing, by divorcing symbolic analysis from material embodiment, has fostered a belief in information as immaterial, but more is at stake in this move to "biology." Notably, Claude Shannon's influential 1936 masters thesis, which showed that relay and switching can be symbolically analyzed (and designed) using Boolean logic, did not rely on an analogy between relays and neurons.[13] In *A Symbolic Analysis of Relay and Switching Circuits*, Shannon develops a means for simplifying and systematizing the development of complex electrical systems. He argues, "Any circuit is represented by a set of equations, the terms of the equations corresponding to the various relays and switches in the circuit." He then goes on to develop a calculus "for manipulating these equations by simple mathematical processes, most of which are similar to ordinary algebraic algorisms."[14] Shannon neither turns to biology nor elaborates on the material details of switches to ground his symbolic analysis. So why should the formal schematic of an automatic stored-memory computer be biologically inflected? And, why does a logical calculus—Boolean, digital logic—necessitate the erasure of the actual functioning of elements, such as vacuum tubes? To respond to these questions, I begin with another: How exactly are analog and digital related in electronic computing?

Nothing but Analog, All the Way Down

According to von Neumann in his 1948 "General and Logical Theory of Automata," a text that intriguingly reverses his initial analogy between vacuum tubes and neurons, the difference between "analogy and digital machines" lies in the ways they produce errors. Analogy machines, von Neumann contends, treat numbers as physical quantities. In order to perform a calculation, they thus find "various natural processes which act on these quantities in the desired way," such as wheel and disk integrators (which lie at the heart of the first computer mice). According to von Neumann, the guiding principle of analogy machines is the classic signal/information-to-noise ratio, a concept Shannon addresses in his *Mathematical Theory of Information*. That is, "the critical question with every analogy procedure is this: How large are the uncontrollable fluctuations of the mechanism that constitute the 'noise,' compared to the significant 'signals' that express the numbers on which the machine operates?"[15] If the calculation to be performed is complex and multistepped, such as the solving of partial differential equations, noise is amplified at every juncture, making it difficult to separate error from answer. Digital machines, in contrast, treat numbers as "aggregates of digits," rather than as physical quantities or signals. Because of this, they are not subject to noise constraints and offer the possibility of absolute precision, although von Neumann points out that round-off errors (now largely addressed by floating-point arithmetic) limit a digital machine's accuracy. Regardless, "the real importance of the digital procedure lies in its ability to reduce the computational noise level to an extent which is completely unobtainable by any other (analogy) procedure."[16]

Crucially, this reduction in noise occurs by ignoring the "analogy" aspect of digital components, for almost every element is a mixture of analogy and digit, as von Neumann acknowledges in "General and Logical Theory of Automata." In opposition to his "First Draft," this later article treats "living organisms as if they were purely digital automata." Responding to objections to this treatment, such as the fact that neurons do not simply work in an all-or-none fashion, he contends:

> In spite of the truth of these observations, it should be remembered that they may represent an improperly rigid critique of the concept of an all-or-none organ. The electromechanical relay, or the vacuum tube when properly used, are undoubtedly all-or-none organs. Indeed, they are the prototypes of such organs. Yet both of them are in reality complicated analogy mechanisms, which upon appropriately adjusted stimulation respond continuously, linearly or non-linearly, and exhibit the phenomena of "breakdown" or "all-or-none" response only under very particular conditions of operation.[17]

The digit, in other words, often treats a quantity as a discrete number, its accuracy resulting from a cut in a signal. The circularity of this passage, in which vacuum tubes are declared prototypes for all-or-none machines, is remarkable. Based on an analogy to computing elements, neurons, which themselves grounded computing elements as digital, are declared digital: an initial analogy is reversed and turned into ontology. At the base of this logic lies a redefinition of analogy itself as a complicated mechanism that operates on continuous quantities, rather than on discrete units.

This redefinition of analog as continuous, still present with us today whenever we refer to film and other media as "analog media," reveals a fundamental ambiguity at the core of what would become known as analog machines: does the analogy take place at the level of the machine architecture or at the level of signal? Analog as model emphasizes analogous differential equations and thus nonobvious analogous effects; analog as continuous buries these likenesses and privileges data over process. According to Thomas D. Truitt and A. E. Rogers in their 1960 *Basics of Analog Computers*:

> The word "analog" (or "analogue") has been used and misused. It has one meaning to some people, and a variety of uses to others. Webster speaks of a thing which maintains "a relation of likeness with another, consisting in the resemblance not of the things themselves, but of two or more attributes, or effects. . . . It is important to recognize that while *analog computer* refers most commonly to this one specific type of analog computer [general purpose d-c electronic analog computer], it can just as well refer to certain mechanical and hydraulic devices, to general purpose a-c electronic computers, and to a variety of special purpose computers. All of these have one characteristic in common—that the components of each computer or device are assembled to permit the computer to perform as a model, or in a manner analogous to some other physical system.[18]

Truitt and Rogers contend that similarities in system behavior, rather than resemblances between individual components, are key. In this sense, analog machines are simulation machines par excellence. Analog computers are based on similar physical relationships between mechanical and electronic systems and emphasize quantities over numbers. That is, the "signal" operated on and the result measured is a physical quantity, such as the intensity of an electrical current, or the rotation of a disk. Importantly, the notion of these machines as "analogy" machines only became apparent after the introduction of what would become digital computers, simulacra par excellence.

Analog to What?

Analog elements, even as they "ground" digital ones such as transistors and neurons, are not simple predecessors to digital computers. Analog and digital machines both thrived in the 1940s through the 1960s. Analog computers were used regularly in nuclear reactors for real-time data processing, as part of real-time control systems, such as flight simulators, and to simulate guided missiles in 3D (they were used to build the intercontinental ballistic missiles, which made the SAGE (Semi-Automatic Ground Environment) air defense system obsolete by the time it was completed).[19] So-called analog computers were popular because of their speed: they could solve problems in parallel, rather than serially (one step at a time), and although digital machines could complete one operation (such as subtraction) much more quickly than analog machines, they were not necessarily faster at complex operations. Early analog machines, as argued earlier, also offered a real-time graphical display that allowed engineers to see immediately how changing a coefficient or variable would alter a problem. Last, the fact that analog computers offered fewer decimal points in their solutions than their digital counterparts was often not important, since the accuracy of the calculation was frequently limited by other factors (measuring input, inadequate equations, etc.) and since early digital computers had significant digit control problems.

Not only were analog computers not viewed or accepted as stepping-stones toward digital ones, but also the division itself between analog and digital electronic computers was not clear. Electronic differential analyzers such as MADDIDA (Magnetic Drum Digital Differential Analyzer), which operated using Boolean algebra and digital electronic circuits, yet treated the signals to be operated as quantities rather than numerical entities, muddied the boundary between analog and digital machines —a boundary that arguably did not then exist. Indeed, analyzers only became analog computers rather than "mechanical mathematical" machines after electronics had displaced electromechanics in the production of discrete and nondiscrete machines.[20]

Electronics arguably marked a "break" between newer and older calculating machines in the 1940s as significant as the difference between digital and "analogy." In the May 1946 press release announcing the ENIAC (Electronic Numerical Integrator and Computer—the first working electronic digital computer), the U.S. War Department introduced it as the first "all-electronic general purpose computer," and underscored its "electronic methods."[21] Electronics marked the ENIAC's difference from both the mechanical "analog" differential analyzer and the "digital" (yet electromechanical) Harvard Automatic Sequence Calculator (Mark 1). In Vannevar Bush's 1945 Franklin Institute article introducing the electromechanical Rockefeller Differential Analyzer (RDA, built in 1942)[22] and in the press releases circulated that year, the RDA is never described by its makers/promoters as an analog machine, but rather as a "machine approach" to mathematics,[23] a "computing machine which marks a significant advance in the field of mechanized mathematics"[24] or, more colloquially, as an "electromechanical giant,"[25] a "tireless ally of science."[26] In response to these publications and to the War Department's announcing the ENIAC, newspapers reported on the machines together, calling them both "Magic Brains"[27] and "Mathematical Robots."[28]

Electronic devices were an important breakthrough because of their speed, and because they were built using nonspecialized labor. Mechanical differential analyzers required trained operators to be present at all times and inadvertently "taught" calculus to its "uneducated" operators. Bush claimed that the integraph (an early electronic version of the differential analyzer) enabled operators/students to cope with difficult mathematical questions by providing “the man who studies it a grasp of the innate meaning of the differential equation.” For such a man, “one part at least of formal mathematics will become a live thing.”[29] Seeing wheel and disk integrators in action makes calculus “live,” moving it from formal writing to actual experience. According to Larry Owens, differential analyzers offered engineering students a graphic way to “think straight in the midst of complexity”—a type of thinking indebted to an engineering “graphical idiom,” which operated as a universal language.

At the core of early analog analyzers lie ordinary differential equations. Similar ordinary differential questions describe seemingly disparate and unrelated electrical, electromechanical, mechanical, and chemical phenomena, all of which can be understood as closed “circuits.” Analog machines, in this sense, work because ordinary differential equations are universal at a large scale, and because Newton’s laws describing force can also describe electrical charge and water capacity.[30] For instance, the mechanical spring circuit represented in figure 4.1 corresponds to the RLC circuit in figure 4.2:

The mechanical spring system corresponds to the following formula:

$m(\mathrm{d}^2x/\mathrm{d}t^2) = \mathrm{F}$ [force] $- kx$ [oscillating force of spring] $- D(\mathrm{d}x/\mathrm{d}t)$
[dissipative force of friction]

The electrical system of figure 4.2 has the following analogous differential equation (see table 4.1 for the corresponding quantities):

$L(\mathrm{d}^2q/\mathrm{d}t^2) = \mathrm{V}$ [voltage] $- 1/Cq$ [oscillating capacitor charge] $- R(\mathrm{d}q/\mathrm{d}t)$
[charge lost over resistor]

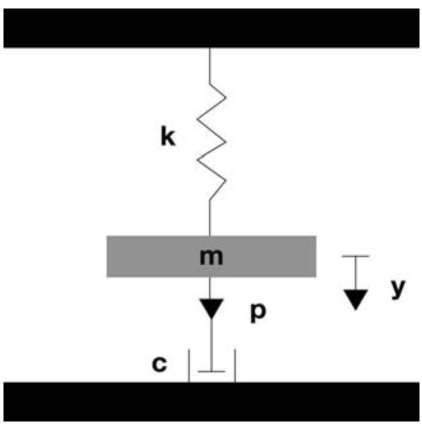


Figure 4.1
Mechanical spring circuit

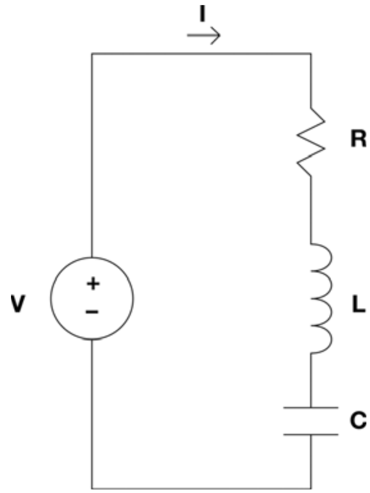


Figure 4.2
RLC circuit

Table 4.1
Analogous entities in the two systems

Mechanical	Electrical
force F	voltage V
mass m	inductance L
friction coefficient D	resistance R
displacement x	charge q
velocity dx/dt	current I
spring coefficient k	reciprocal of capacity $1/C$

All these equations could be put in the form

$D^{n-1}y/dx^{n-1} = \int d^n y/dx^n dx.$

For the mechanical spring system, this would be

dx/dt [velocity] $= (1/m) \int(F - kx - D(dx/dt))dt.$

These equations are not usually solvable using normal analytic methods, but can be solved using numerical methods (desk calculators generally produced tables of solutions to differential equations before the popularization of machinic computers). MIT's differential analyzers employed a wheel and disc integrator to solve these differential

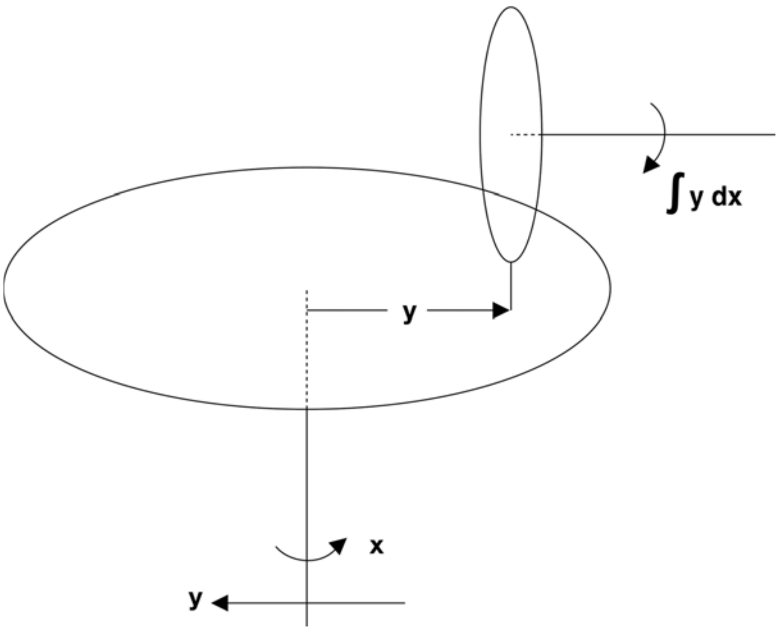


Figure 4.3
Schematic of a basic wheel and disk integrator

equations mechanically, using feedback to solve for values, which appeared on both sides of the equation sign. Figure 4.3 gives the basic design and principle of the integrator.

As figure 4.4 makes clear, the distance *y* is not a static value, but rather a function given determined by the rotation of another shaft.

So that

$$W = k\int_{v1}^{v} U dV.$$

To schematically represent the various operations, Bush used the following symbols (see figure 4.5):

So, using the equation $d^2y/dx^2 = f(x)$, in which case $f(x)$ is known in order to solve for *y*, one would build the setup outlined in figure 4.6.

Crucially, the differential analyzer employed "generative" functions—that is, the output could feed into itself. It could thus solve for variables on both sides of the equation. For instance, consider the solution for $d^2y/dx^2 = f(y)$, which is shown in figure 4.7.

These generative functions mark a fundamental difference between digital machines, which solve problems step by step, and analog machines.

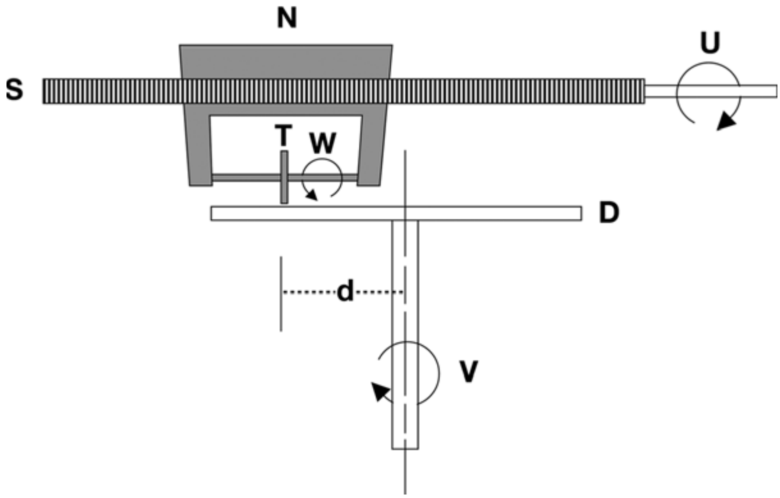


Figure 4.4
Integrator geometry

Because of this mechanical yet "live," analogous relationship, analog machines have generally been conceptualized as more transparent and intuitive than digital ones. Samuel Caldwell, director of MIT's Center of Analysis, stated, "There is a vividness and directness of meaning of the electrical and mechanical processes involved . . . [whereas] a Digital Electronic computer is bound to be a somewhat abstract affair, in which the actual computational processes are fairly deeply submerged."[31] Historian Paul Nyce has argued this mechanical mirroring made the move from analogy to essence or ontology difficult: one always dealt with—made visible—two analogous situations, rather than a universal solution. Nyce contends that analog devices

> belong to a long tradition of scientific instruments, starting in the seventeenth century, that "made visible what could not be seen" . . . Unlike most scientific instruments, however, analog devices supported both understanding (literally by measurement and number, like an astrolabe) and investigation for they, like an orrery, were "models" of phenomena. . . . What also made them persuasive is that they were both statements about and direct imitations of the things they represented. Mimesis is "hidden" or absent in digital machines: analog machines represent phenomena vividly and directly."[32]

Intriguingly, direct representation—or more accurately correspondence—makes analog machines live, vivid, and direct. It is a representation that always is tethered to another "source," which it does not try to hide. The differential analyzer was not, as the digital computer would be, amenable to notions of "universal" disembodied information. The differential analyzer simulated other phenomena, whereas digital computers, by

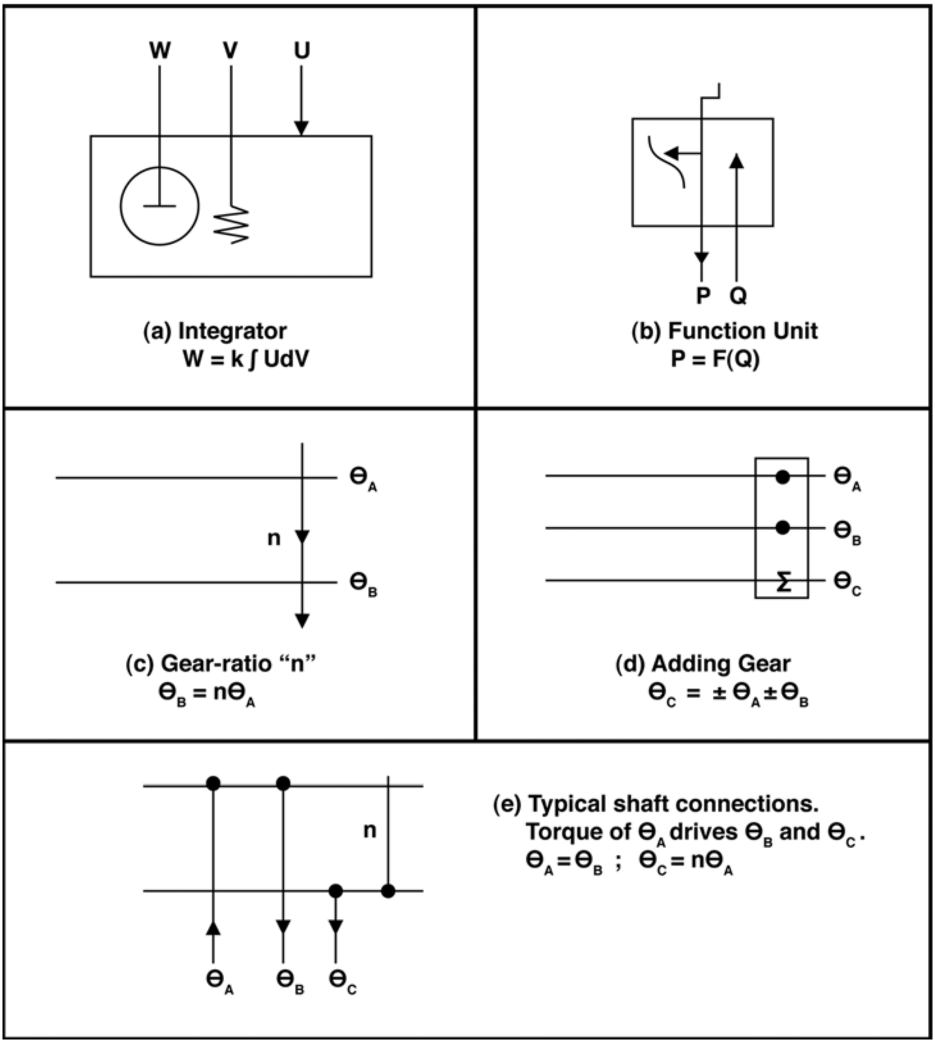


Figure 4.5
Symbols used in connection diagrams for a differential analyzer

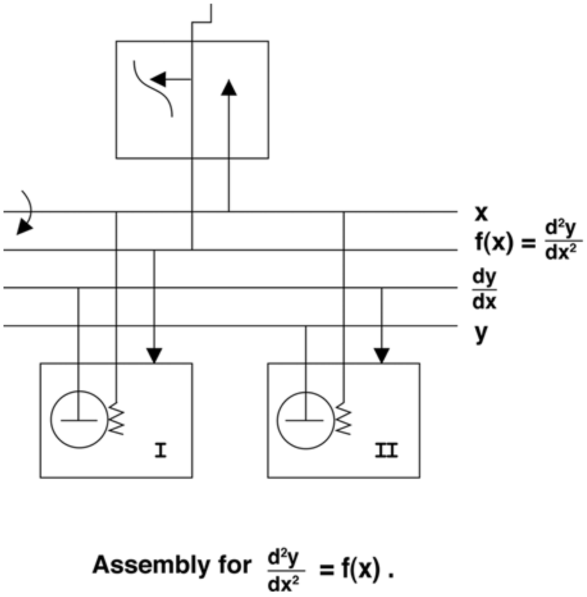


Figure 4.6
$d^2y/dx^2 = f(x)$

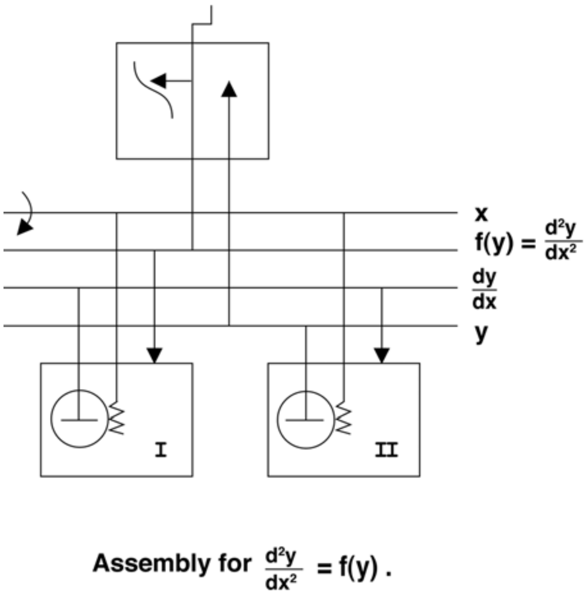


Figure 4.7
$d^2y/dx^2 = f(y)$

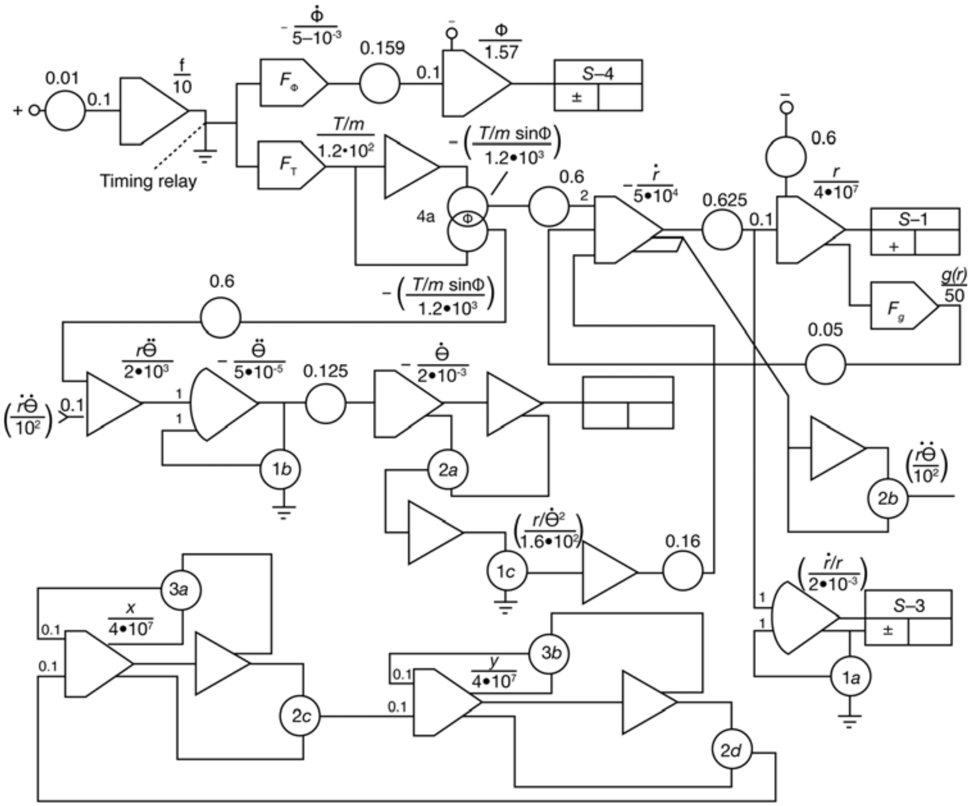
hiding mimesis, could simulate any other machine. That is, while both digital and analog computers depend on analogy, digital computers, through their analogy to the human nervous system (which we will see stemmed from a prior analogy between neurons and Turing machines), simulate other computing machines using numerical methods, rather than recreating specific mechanical/physical situations. They move us from "artificial representation" or mechanical analysis (description) to simulacra or "information" (prescription). They move us from solving a problem by defining its parameters to solving it by laying out a procedure to be followed step by step. Depending on one's perspective, analog computers either offer a more direct, "intuitive," and, according to Vannevar Bush, "soul-satisfying" way of solving differential equations or they are imprecise and noisy devices, which add extra steps—the translation of real numbers into physical entities.[33] The first, the engineer's perspective, views computers as models and differential equations as approximations of real physical processes; the second, the mathematician's perspective, treats equations as predictors, rather than descriptors of physical systems—the computer becomes a simulacrum, rather than a simulation.

To be clear, though, analog machines did not simply operate via analogy; again, the notion that they operated through analogy would only be apparent later. They dealt with "signals," from which the notion and the theory of information would emerge, and further Vannevar Bush, as an electrical engineer, considered electricity to be a universal principle.[34] As well, to return to the question of electronics, all analog machines are not large, "intuitively understood," "live" mechanical devices. The electronic machines of the 1950s and 60s differed significantly from their mechanical predecessors. We thus need to be careful not to base arguments about analog machines as a whole on Vannevar Bush's early machines.[35] Indeed, Bush and Caldwell argued that one benefit of the electromechanical RDA was the fact that a trained operator was not necessary. As they explained, the user no longer had to "keep up "with the machine.[36] Op-amps as integrators, or even multipliers, were not "seeable" and graspable in the same fashion as wheel and disks.[37] Last, analog computational structures do not have to coincide perfectly with the problem to be solved: one can reuse an integrator in the same way that one can reuse an adder.

The move to electronics not only deskilled operators, it also made computers mass producible. The mechanical differential analyzers were steeped in the "odor" and the specialized labor of the machine lab, and they used special cams hand-crafted by highly skilled mechanics (the University of Pennsylvania Moore School Differential Analyzer was a WPA [Works Progress Administration] project, designed to employ mechanics). B. Holbrook, who worked at Bell Labs, argued that wire-wound potentiometers "offered the possibility of getting a completely new and relatively trainable type of labor into the manufacture of these things instead of the very high precision mechanics that were necessary by using the prior method."[38] Electronic analog and digital computers used mass-produced vacuum tubes and later transistors. Thus, both electronic analog and digital "machines" participate in Fordist logic: they automate calculation and production and make invisible the mathematics or calculations on which they rely.

Digital machines, however, are more profoundly Fordist than analog ones. The War Department ENIAC press release states that the ENIAC will eliminate expensive design processes: "Many electrical manufacturing firms, for instance, spend many thousands of dollars yearly in building 'analogy' circuits when designing equipment."[39] Most significant, they are more Fordist because their programming breaks down problems into simple, repeatable discrete steps. It is in programming, or to be more precise, programming in opposition to coding, that analog and digital machines most differ. Douglass Hartree, in his 1949 *Calculating Instruments and Machines*, reserves the terms *programming* and *coding* for digital machines, even though the RDA used tapes to specify the required interconnections between the various units, the values of ratios for the gearboxes, and the initial displacements of the integrators.[40] These tapes, unlike ones used for digital electronic computers, did not contain instructions necessary for sequencing a calculation; like von Neumann, Hartree describes programming as the "drawing up [of a] schedule of [the] sequence of individual operations required to carry out the calculation," and coding as the "process of translating operations into instructions in the particular form in which they are read by the machine."[41] Digital and analog electronic programming both retained the iconographic language of the differential analyzers, and in this sense were both grounded in mechanical methods or in their simulation. However, whereas digital flow charts produce a sequence of individual operations, analog programming produces a "circuit" diagram of systematic relations (see figure 4.8). These differences in programming also point toward key internal differences in representation, namely numbers versus quantities. Coded digital machines are much easier to follow. At a certain level then, analog machines (especially mechanical ones) were not simply more visual or transparent, but rather more complicated.

This complexity made it unlikely that analog computers could spawn or support code as logo*s*. Code as logos—code as the machine—is intimately linked to digital design, which enables a strict step-by-step procedure that neatly translates time into space. Although later it would threaten to reduce all hardware to memory devices in the minds of most of its users, code as logos depended on a certain "hard" digital logic. This logic turns neurons and vacuum tubes themselves into logos and produces an insatiable need for memory, understood as regenerative circuits. This logic again stems from "biology," or, rather, from technologically enhanced biology: cybernetics.



Scaled computer diagram for simulation of satellite system

Figure 4.8
An analog program diagram, based on an image from Albert S. Jackson, *Analog Computation* (New York: McGraw-Hill Book Company, 1960), 266

In the Beginning Was Logos (Again)

In "A Logical Calculus of the Ideas Immanent in Nervous Activity," McCulloch and Pitts seek to explain the operation of the brain in logical terms. This paper is part of McCulloch's larger project of "experimental epistemology," his effort to explain "how we know what we know . . . in terms of the physics and chemistry, the anatomy and physiology, of the biological system."[42] This experimental epistemology did not shun theory, but rather sought to weave together philosophy and neurophysiology. At its heart lies the equation of "the 'all-or-none' character of nervous activity" with propositional logic. It reduces a neuronal action to a statement capable of being true or false, "to a proposition which proposed its adequate stimulus."[43] This equation once more conflates word with action: in this particular case, the firing of a neuron with the proposition that "made" it fire. (Not surprisingly, McCulloch describes his examination of the human mind as a "quest of the Logos.")[44] This equation also concretizes the mind and ideas: "With the determination of the net," McCulloch and Pitts write, "the unknowable object of knowledge, the 'thing itself,' ceases to be unknowable."[45]

As the quotations around "all-or-none" imply, this description is a simplification, one coupled with assumptions such as: "a certain fixed number of synapses must be excited within a period of latent addition in order to excite a neuron at any time, and this number is independent of previous activity and position on the neuron."[46] Despite this, they argue that the all-or-none behavior of neurons makes them the fundamental psychic units or "psychons," which can be compounded "to produce the equivalents of more complicated propositions" in a causal manner.[47] Indeed, the goals of McCulloch and Pitts's logical calculus are to calculate the behavior of any neural net and to find a neural net that will behave in a specified way.[48] Remarkably, their method to "know the unknowable" not only simplifies nervous activity, it also does not engage the actual means by which inhibition or excitation occurs. This is because their method considers circuits equivalent if their result—their perceived behavior—is the same (as I explain later, this was crucial to cybernetic memory). Further, they erase actual alterations that occur during facilitation and extinction (antecedent activity temporarily alters responsiveness to subsequent stimulations of same part of the net) and learning (activities concurrent at some previous time alters the net permanently) via fictitious nets composed of ideal neurons whose connections and thresholds are unaltered.[49] Even though they state that formal equivalence does not equal factual explanation, they also insist that the differences between actual and idealized action do not affect the conclusions that follow from their formal treatment, namely the discovery/generation of a logical calculus of neurons.

Importantly, this logic of equivalence between neural nets and propositional logic was grounded, for McCulloch, in the nature of numbers themselves. In "What Is a Number, that Man May Know It, and a Man, that He May Know a Number?," he draws from David Hume to argue that only numbers truly can be equal. McCulloch's definition of numbers is Bertrand Russell's, "a number is the class of all those classes that can be put into one-to-one correspondence to it."[50] McCulloch's logical calculus, in other words, could only be digital with 1s and 0s corresponding to true and false. McCulloch later made this explicit, in his 1951 "Why the Mind Is in the Head," distinguishing the nervous system from sense organs in terms of digital versus analog. "In so-called logical, or digital contrivances," he writes, "a number to be represented is replaced by a number of things—as we may tally grain in a barn by dropping a pebble in a jug for each sheaf . . . the nervous system is par excellence a logical machine."[51] To McCulloch, logical equals digital because they both rely on numbers. Although analog machines also imply and are based on one-to-one models, McCulloch, focusing on signals rather than on the machine, claims, "in so-called analogical contrivances a quantity of something, say a voltage or a distance, is replaced by a number of whatnots or conversely, quantity replaces the number. Sense organs and effectors are analogical."[52] In this schema, analog to digital conversion takes place at the level of data—the difference in machine technology is completely erased through a logic of equivalence.

By calling the cortex a digital machine, McCulloch sought to displace the then popular theory of the mind as functioning mimetically. According to Seymour Papert, McCulloch liberated the theory of perception from "the idea that there must be in the brain some sort of genetically faithful representation of the outside world."[53] This is most clearly seen in his 1959 "What the Frog's Eye Tells the Frog's Brain," (an article with J. Y. Lettvin, H. R. Maturana, and W. H. Pitts). In it, they argue that because a frog's eye does not transmit a copy of what it sees but rather detects certain patterns of light and their changes in time, the "eye speaks to the brain in a language already highly organized and interpreted, instead of transmitting some more or less accurate copy."[54] Even earlier, though, and before von Neumann's preliminary draft, the cortex for McCulloch was a Turing machine. In "A Logical Calculus," McCulloch and Pitts state, "Every net, if furnished with a tape, scanners connected to afferents and suitable efferents to perform necessary motor-operations, can compute only such numbers as can a Turing machine."[55] Neural nets are inspired by and aspire to be Turing machines.[56] Von Neumann's use of McCulloch and Pitts's analysis is thus an odd and circular way of linking stored-memory digital computers to computing machines—once more, an over-determined discovery of a linkage between biology and computer technology, yet another turn of the double helix (before, of course, there was a double helix).

This linkage not only establishes a common formal logic, it also enables the emergence of computer "memory." Moving away from ideas of field-based, analogical notions of memory, McCulloch's neural nets produce transitory memories and ideas through circular loops. Drawing from Wiener's definition of information as order (negative entropy), McCulloch argues that ideas are information: they are regularities or invariants that conserve themselves as other things transform.[57] McCulloch contentiously claims that this stability is produced by reverberating "positive-feedback" circuits, that is, transitory memory (reverberatory memory cannot survive a "shut down," such as a deep sleep or narcosis).[58] These reverberatory circuits, though, even as they enable memory, also render "reference indefinite as to time past,"[59] for what is retained is the memory, not all the events that led to that memory. In this sense, they threaten to become "eternal ideas," separated from context. This separation, combined with the fact that the neural nets can specify the next but not the previous state, means that "our knowledge of the world, including ourselves, is incomplete as to space and indefinite as to time."[60] Causality runs only one way: one cannot decisively "reverse engineer" a neural net's prior state.

This emergence of memory is thus, as Bowker notes, also a destruction of memory. Thinking through cybernetician Ross Ashby's claim that "memory is a metaphor needed by a 'handicapped' observer who cannot see a complete system," Bowker writes, "The theme of the destruction of memory is a complex one. It is not that past knowledge is not needed; indeed, it most certainly is in order to make sense of current actions. However, a *conscious* holding of the past in mind was not needed: the actant under consideration—a dog, a person, a computer—had been made sufficiently different that, first, past knowledge was by definition retained and sorted and, second, only useful past knowledge survived."[61] What is truly remarkable is that this destruction of memory has spawned the seemingly insatiable need for computer memory. Memories are rendered into context-free circuits freed from memory, circuits that are necessary to the operation of the animal/machine.

Although the past may not be determinable from the present, memories—as context-free invariant patterns—ground our ability to predict the future. This prediction—causality—according to McCulloch (drawing from Hume) is only a "suspicion"[62] that there is "some law compelling the world to act hereafter as it did of yore."[63] Like those of ideas, these predictive circuits persist. Indeed, McCulloch argues, "the earmark of every predictive circuit is that if it has operated long uniformly it will persist in activity, or overshoot; otherwise it could not project regularities from the known past upon the unknown future."[64] The endurance of these circuits, however, threatens closure, threatens to make the unknown imperceptible, something that McCulloch "as a scientist . . . dread[s] most, for as our memories become stored, we become creatures of our yesterdays—mere has-beens in a changing world. This leaves no room for learning."[65] Memory, then, which enables a certain causality as well as an uncertainty as to time and place, threatens to overwhelm the system, creating networks that crowd out the new. A neural circuit, if it persists—programmability—makes prediction possible. It, however, also puts in jeopardy what for McCulloch is most interesting and vital about humanity: the ability to learn and adapt to the unknown, that is, the future as future.

This notion of memory as circuit/signal underscores McCulloch's difference from cognitive psychology, which, following developments in computer technology, would consider the brain hardware and the mind software.[66] In McCulloch's system, the mind and body are intimately intertwined, with the mind becoming less "ghostly"—more concrete—perhaps paradoxically by becoming signal.[67] Signals bridge mind and brain because they have a double nature; they are both physical events and symbolic values.[68] They are both statement and result. The logic of computers as logos stems from the disciplining, the axiomatizing, of hardware. This in turn "solidifies" instructions into things in and of themselves. Notably, McCulloch in his later work did address software, or programs, but referred to them as instructions to be operated on by data in memory, rather than as stored themselves in memory.[69] Instructions, in other words, did not drive the system—the logic, the logos, happened at the level of firing neurons.

Thus, by turning to McCulloch and Pitts rather than to Shannon, von Neumann gains a particular type of abstraction or logical calculus: an axiomatic abstraction and schematic design that greatly simplifies the behavior of its base components. Von Neumann also gains a parallel to the human nervous system, key to his later work on "general automata." Last, he "gains" the concept of memory—a concept that he would fundamentally alter by asserting the existence of biological organs not known to exist. Through this hypothetical "memory organ," and his discussion of the relationship between orders and data, his model would profoundly affect the development of cognitive science and artificial intelligence (AI) and life (AL). Through this memory organ, von Neumann would erase the difference between storage and memory, and also open up a different relationship between man and machine, one that would incorporate instructions—as a form of heredity—into the machine, making software fundamental. If word (as description) becomes event in McCulloch and Pitts's theory, in von Neumann's theory event once again becomes word, word becomes instruction.

Memories to Keep in Mind

Von Neumann's work with natural and artificial automata in general reverses the arrow of the analogy established in "First Draft." Rather than explaining computers in terms of the human nervous system, he elucidates the brain and its functioning in terms of computational processes. This is most clear in von Neumann's discussion of memory, which he considered to be a "much more critical and much more open" issue than logical processing.[70] In computer systems, memory was the bottleneck, for the limitations of memory on the machine created an "abnormal economy," in which the computer is forced to store all the information it needs to solve a problem on the equivalent of one page.[71]

The term *memory organ* clearly borrows from biology. This borrowing, however, was not necessary. Prior to "First Draft," mechanisms designed to store numbers and functions necessary for computing were called storage devices or "the store," following Babbage's terminology. J. Presper Eckert's 1944 "Disclosure of Magnetic Calculating Machine," used as evidence in the patent trial, refers concretely to the disks or tapes used to store data; his 1946 patent application, in contrast, employs the term *electrical memory*. This movement from storage to memory lies at the heart of the computer as archive, the computer as saving us from the past, from repetition through repetition.

Computer storage devices as memory is no simple metaphor, since it asserts the existence of an undiscovered biological organ. Although von Neumann initially viewed memory as comprising afferent neurons, he soon changed his mind, based on his own experience with computers, in particular with the number of vacuum tubes needed to create the types of reverberatory circuits McCulloch and Pitts described. In a reverse move, he postulated human memory as something unknown but logically necessary, making clear that his first analogy was based on a leap of faith. In *The Computer and the Brain*, written ten years after "First Draft," von Neumann writes, "the presence of a memory—or, not improbably, of several memories—within the nervous system is a matter of surmise and postulation, but one that all our experience with artificial automata suggests and confirms." Von Neumann goes on to emphasize our ignorance regarding this memory:

> It is just as well to admit right at the start that all physical assertions about the nature, embodiment, and location of [human memory] are equally hypothetical. We do not know where in the physically viewed nervous system a memory resides; we do not know whether it is a separate organ or a collection of specific parts of other already known organs, etc. It may well be residing in a system of specific nerves, which would then have to be a rather large system. It may well have something to do with the genetic mechanism of the body. We are as ignorant of its nature and position as were the Greeks, who suspected the location of the mind in the diaphragm. The only thing we know is that it must be a rather large-capacity memory, and that it is hard to see how a complicated automaton like the human nervous system could do without one.[72]

This passage reveals how quickly the computer moved from a system modeled on ideal neurons to a concrete model for more complex biological phenomena. This statement, which seems to be so careful and qualified—we basically do not know what the memory is or where it resides—at the same time asserts the existence of a memory organ or set of organs based on an analogy to computers: "The only thing we know is that it must be a rather large-capacity memory, and that it is hard to see how a complicated automaton like the human nervous system could do without one." This guess regarding capacity assumes that the brain functions digitally, that it stores information as bits, which are then processed by the brain, rather than functioning more continuously in a "field-based" manner. Again, this assumption was by no means accepted whole-heartedly by biologists. Dr. Lashley, among others, responded to von Neumann's difficulty with neuronal capacity by arguing that the memory was more dynamic rather than static and that "the memory trace is the capacity of many neurons to work together in certain permutations."[73]

Neurons as switching elements drive von Neumann's "logical" guess regarding memory capacity, as well as his confusion over its location:

> In the human organism, we know that the switching part is composed of nerve cells, and we know a certain amount about their functioning. As to the memory organs, we haven't the faintest idea where or what they are. We know that the memory requirements of the human organism are large, but on the basis of any experience that one has with the subject, it's not likely that the memory sits in the nervous system, and it's really very obscure what sort of thing it is.[74]

Digital switching devices, based on the reduction of all processes to true/false propositions, insatiably demand memoryless memory. As von Neumann explains in "First Draft," the need for memory increases as problems are broken down into long and complicated sequences of operations (described in chapter 1 of this book by Bartik and Holberton). Digital computation needs to store and have access to intermediate values, instructions, specific functions, initial conditions and boundary conditions, etc. Prior to the EDVAC, these were stored in an outside recording medium such as a stack of paper cards. The EDVAC was to increase the speed of calculation by putting some of those values inside the memory organ, making porous the boundaries of the machine. Memory instituted "*a prosthesis of the inside*."[75] Memory was not simply sequestered in the "organ"; it also bled into the central arithmetic unit, which, like every unit in the system, needed to store numbers in order to work.

To contain or localize memory, von Neumann organized it hierarchically: there were to be many memory organs, defined by access time rather than content. For instance, in the 1946 work "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," von Neumann and colleagues divide memory into two conceptual forms—numbers and orders, which can be stored in the same organ if instructions are reduced numbers—and into two types—primary and secondary.[76] The primary memory consists of registers, made of flip-flops or trigger circuits, which need to be accessed quickly and ideally randomly. Primary memory, however, is very expensive and cumbersome. A secondary memory or storage medium supplements the first, holding values needed in blocks for a calculation. Besides being able to store information for periods of time, such a memory needs to be controllable automatically (without the help of a person), easily accessed by the machine, and preferably rewriteable. Interestingly, the devices listed as possible secondary memories are other forms of media: for instance, teletype tapes, magnetic wire or tapes, and movie film. (The primary media was also another medium—the Selectron was a vacuum tube similar to one used for television.)[77] This gives a new resonance to McLuhan's assertion that new media do not make preexisting media obsolete but merely change their use.[78] Von Neumann and colleagues also outlined a third form of memory, "dead storage," which is an extension of secondary memory, since it is not initially integrated with the machine. Not surprisingly, input and output devices eventually become part of "dead storage." As von Neumann argues later in *The Computer and the Brain*, "the very last stage of any memory hierarchy is necessarily the outside world, that is, the outside world as far as the machine is concerned, i.e. that part of it with which the machine can directly communicate, in other words the input and the output organs of the machine."[79] In this last step, the borders of the organism and the machine explode. Rather than memory comprising an image of the world in the mind, memory comprises the whole world itself as it becomes "dead."

This last step renders the world dead by conflating memory—which is traditionally and initially regenerative and degenerative—with other more stable forms of media such as paper storage, a comparison that is still with us today at the level of both memory (files) and interface (pages and documents). This conflation both relied on and extended neurophysiological notions of memory as a trace or inscription, like the grooves of a gramophone record. McCulloch, for instance, in 1951, in response to objections posed by von Neumann over memory as reverberatory circuits, outlined a hierarchical memory system that resonated with von Neumann's schema. There are first temporary reverberations, and second, nervous nets that alter with use (central to conditioned behaviors). The third type of memory, which he sees as an informational bottleneck, however, leaves him unhappily stumped; he is at a loss to describe its location and its operation:

> I don't see how we can tell where we have to look as yet, because in many of the experiments in which there are lesions made in brains, we have had large amounts of territory removed. However, usually we fail to destroy most fixed memories: therefore, we cannot today locate the filing cabinets. I think that sooner or later answers to the question of those filing cabinets, or whatever it is on which is printed "photographic records" and what not, will have to be found.[80]

The term *filing cabinet* is drawn from von Neumann's own terminology. In his response to McCulluch's paper, von Neumann, perhaps informed by psychoanalytical arguments that memories never die (one of von Neumann's uncles introduced psychoanalysis to Hungary and von Neumann apparently loved to analyze jokes) or by his personal experience (he allegedly had a photographic memory and could recall conversations word for word), presents the following "negative" and not entirely "cogent" argument against memory as residing in the neurons:

> There is a good deal of evidence that memory is static, unerasable, resulting from an irreversible change. (This is of course the very opposite of a "reverberating," dynamic, erasable memory.) Isn't there some physical evidence for this? If this is correct, then no memory, once acquired, can be truly forgotten. Once a memory-storage place is occupied, it is occupied forever, the memory capacity that it represents is lost; it will never be possible to store anything else there. What appears as forgetting is then not true forgetting, but merely the removal of that particular memory-storage region from a condition of rapid and easy availability to one of lower availability. It is not like the destruction of a system of files, but rather like the removal of a filing cabinet into the cellar. Indeed, this process in many cases seems to be reversible. Various situations may bring the "filing cabinet" up from the "cellar" and make it rapidly and easily available again.[81]

Von Neumann's "negative argument" relies on files and the human mind as the owner/manipulator—or, to return to Cornelia Vismann's argument outlined in chapter 2, chancellor—of files. It also depicts the human brain as surprisingly nonplastic: easily used up and unerased, hence once more the need for great storage. It also moves away from memory as based on erasable "regenerative" traces toward fantasies of traces that do not fade: immortality within the mortal machine.[82] This is a far cry from Vannevar Bush's description of the human mind in chapter 2 as fundamentally ephemeral and prone to forgetting. The digital paradoxically produces memory as storage, in part because logical algorithms need to read and write values. An entire process can fail if one variable is erased.

Memory as storage also allows von Neumann to describe genes as a form of human memory. In *The Computer and the Brain*, he writes, "another form of memory, which is obviously present, is the genetic part of the body; the chromosomes and their constituent genes are clearly memory elements which by their state affect, and to a certain extent determine, the functioning of the entire system."[83] With this move toward genes as memory—necessary for his theory of self-reproducing formula—neurons would not stand in for words (true or false propositions), but words (instructions) would come to stand in for neurons.

Descriptions that Can

The deed is everything, the Glory naught.
—*Faust*, Part II

According to William Poundstone, the last anecdote of von Neumann's "total recall" concerns his last days, when he lay dying of cancer at Walter Reed Hospital, a cancer caused by his work on nuclear weapons (the drive for nuclear weapons also powered the development of digital electronic computers; American computers and neoliberalism are both reactions to Nazism).[84] His brother Michael read *Faust* in the original German to von Neumann and, "as Michael would pause to turn the page, von Neumann would rattle off the next few lines from memory."[85] Converting to Catholicism before his death, von Neumann was deeply influenced by the work of Goethe, *Faust* in particular. Said his brother Nicholas, "We studied *Faust* in school very thoroughly, both parts, in original and in Hungarian translation. And we discussed it for years and rereading it occasionally thereafter, throughout our respective lifetimes."[86] One of the three passages Nicholas describes as particularly important to his brother was Faust's grappling with logos: "Faust's monologue at the opening of the First Part: 'In the beginning was the Act,' and the corresponding statement in Part II: 'The deed is everything, the Glory naught.' This we discussed in the context of the redeeming value of action."[87] According to Nicholas, this passage led "ultimately to John's views emphasizing the redeeming value of practical applications in his profession."[88] John von Neumann as an unredeemed (although not yet fallen) Faust.

This passage, however, has other resonances, intersecting with the question of logos weaving through this book. Faust, seeking to translate the Bible into German pauses over "in the beginning was the Word":

I'm stuck already! I must change that; how?
Is then "the word" so great and high a thing?
There is some other rendering,
Which with the spirit's guidance I must find.
We read: "In the beginning was the Mind."
Before you write this first phrase, think again;
Good sense eludes the overhasty pen.
Does "mind" set worlds on their creative course?
It means: "In the beginning was the Force."
So it should be—but as I write this too,
Some instinct warns me that it will not do.
The spirit speaks! I see how it must read,
And boldly write: "In the beginning was the Deed!"[89]

Faust, after a failed encounter with a spirit he conjured but cannot control, replaces Word with Deed, which, rather than Word, Force, or Mind, creates and rules the hour. Ironically, Faust, of course, is later saved by the Word—a technicality regarding his statement of satisfaction. Regardless, this substitution of Word with Deed sums up von Neumann's axiomatic approach to automata and his attraction to McCulloch and Pitts's work. It also leads him to conceive of memory as storage: as a full presence that does not fade, even though it can be misplaced. What is intriguing, again, is that this notion of a full presence stems from a bureaucratic metaphor: filing cabinets in the basement. This reconceptualization of human memory bizarrely offers immortality through "dead" storage: information as undead.

McCulloch and Pitts's methodology again depends on axiomatizing idealized neurons, where, according to von Neumann, "axiomatizing the behavior of the elements means this: We assume that the elements have certain well-defined, outside, functional characteristics; that is, they are to be treated as 'black boxes.' They are viewed as automatisms, the inner structure of which need not be disclosed, but which are assumed to react to certain unambiguously defined stimuli, by certain unambiguously defined responses."[90] This controversial axiomatization, which von Neumann would employ later in his theory of self-reproducing automata, reduces all neuronal activities to true/false statements.[91] Neurons follow a propositional logic. Von Neumann contends that this axiomatizing and the subsequent logical calculus it allows means that McCulloch and Pitts have proven that "any functioning . . . which can be defined at all logically, strictly, and unambiguously in a finite number of words can also be realized by such a formal neural network. . . . It proves that anything that can be exhaustively and unambiguously described, anything that can be completely and unambiguously put into words, is ipso facto realizable by a suitable finite neural network."[92] Words that describe objects, in other words, can be replaced by mechanisms that act, and all objects and concepts, according to von Neumann, can be placed in this chain of substitution. "There is no doubt," he asserts, "that any special phase of any conceivable form of behavior can be described 'completely and unambiguously' in words. This description may be lengthy, but it is always possible. To deny it would amount to adhering to a form of logical mysticism which is surely far from most of us."[93] This does not mean, however, that such a description is simple; indeed, von Neumann stresses that McCulloch and Pitts's theorizing is important for its reverse meaning: "there is a good deal in formal logics to indicate that the description of the functions of an automaton is simpler than the automaton itself, as long as the automaton is not very complicated, but that when you get to high complications, the actual object is simpler than the literary description."[94]

This notion of an actual object is not outside of language, even if it is outside "literary description," for, to von Neumann, producing an object and describing how to build it were equivalent. For instance, he argues that the best way to describe a visual analogy may be to describe the connections of the visual brain.[95] According to this logic, the instructions to construct a machine can substitute for the machine itself, to the extent that it can produce all the behaviors of the machine.

This logic is most clear in von Neumann's earliest model of self-reproduction, which Arthur Burks later dubbed a "robot" or "kinematic" model.[96] In this model, "constructing automata" *A* are placed in a "reservoir in which all elementary components in large numbers are floating."[97] Automaton *A* "when furnished the description of [an] other automaton in terms of appropriate functions will construct that entity." This description "will be called an instruction and denoted by a letter *I*. . . . All [*A*s] have a place for an instruction *I*."[98] In this system, instruction drives construction. In addition to automata *A*, there are also automata *B*, which can copy any instruction *I* given to them. The decisive step, von Neumann argues, is the following instruction to the reader about embedding instructions:

> Combine the automata *A* and *B* with each other, and with a control mechanism *C* which does the following. Let *A* be furnished with an instruction *I*. . . . Then *C* will first cause *A* to construct the automaton, which is described by this instruction I. Next *C* will cause *B* to copy the instruction *I* referred to above, and insert the copy into the automaton referred to above, which has just been constructed by *A*. Finally, *C* will separate this construction from the system $A + B + C$ and "turn it loose" as an independent entity.[99]

This independent entity is to be called *D*. Von Neumann then argues, "In order to function, the aggregate $D = A + B + C$ must be furnished with an instruction *I*, as described above. This instruction, as pointed out above, has to be inserted into *A*. Now form an instruction I_D, which describes this automaton D, and insert I_D into A within D. Call the aggregate which now results E. E is clearly self-reproductive."[100] This instruction I_D (which nicely resonates with ID and id), he claims, is roughly equivalent to a gene. He also contends that B "performs the fundamental act of reproduction, the duplication of the genetic material, which is clearly the fundamental operation in the multiplication of living cells." This analogy fails, however, because "the natural gene does probably not contain a complete description of the object whose construction its presence stimulates. It probably contains only general pointers, general cues."[101] Thus, the memory of the system—here postulated as a more vibrant form of memory than "paper tape"—becomes the means by which the automaton can self-reproduce.[102]

This description is amazing for several reasons. In it, von Neumann transforms McCulloch and Pitts's schematic neural networks, in which there is no separation of software from hardware, into the basis for code as logos for the instructions replace the machine. What becomes crucial, in other words, and encapsulates the very being of the machine, are the instructions needed to construct it. Furthermore, and inseparable from the translation of event into instruction, this description—as a set of instructions itself—contains a bizarre, almost mystical, address. For, when von Neumann says, "Now form an instruction I_D, which describes this automaton D, and insert I_D into A within D," or "Combine the automata A and B with each other, and with a control mechanism C," who will do this forming and combining; who will perform these crucial steps and how? What mystical force will respond to this call? Like Faust before Mephistopheles arrives, are we to incant spells to create spirits? The transformation of description into instruction leaves open the question: who will do this? Who will create the magical description that goes inside? Remarkably, this call makes clear the fact that humans are indistinguishable from automata, something that bases von Neumann's game theory as well.

Games and Universes

This replacement of descriptions by instructions (or choices among instructions) also grounds von Neumann's work in game theory, which corresponds to his work on automata in many ways, as Arthur Burks has pointed out. "There is a striking parallel," Burks writes, "between von Neumann's proposed automata theory and his theory of games. Economic systems are natural competitive systems; games are artificial competitive systems. The theory of games contains the mathematics common to both kinds of competitive systems, just as automata theory contains the mathematics common to both natural and artificial automata."[103] This comparison, however, not only occurs at the level of mathematics or mathematization, but also at the level of heuristics, descriptions, and strategies. Game theory, which has been a key tool of neoliberal economic theory, seeks to understand the problem of exchange through the perspective of a "game of strategy," in which participants create strategies in response to others' moves, the rules of the game, and (objective) probabilities.[104] Similar to von Neumann's "First Draft," von Neumann and Oskar Morgenstern's 1944 *Theory of Games and Economic Behavior* (their preliminary discussion of game theory) serves as a *heuristic*, a "phase of transition from unmathematical plausibility considerations to the formal procedure of mathematics."[105] Also like his theory of automata, and indeed like most of von Neumann's mathematical work, game theory is based on an axiomatic method. Most importantly, von Neumann and Morgenstern introduce the notion of *strategy* to replace or simplify detailed description. Describing the process of giving an exact description of what comprises a game, they write, "we reach—in several successive steps—a rather complicated but exhaustive and mathematically precise scheme." Their key move is "to replace the general scheme by a vastly simpler one, which is nevertheless equivalent to it. Besides, the mathematical device which permits this simplification is also of an immediate significance for our problem: It is "the introduction of the exact concept of a strategy."[106] A strategy is a complete plan that "specifies what choices [the player] will make in every possible situation, for every possible actual information which he may possess at that moment in conformity with the pattern of information which the rules of the game provide for him for that case."[107] This replacement of a complete description with a strategy is not analogous to the replacement of machine code with a higher-level programming language, or what von Neumann calls "short code." This "equivalence" is not based on a simplification through the creation of a language that reduces several events into one statement, but rather on a fundamental transformation of a step-by-step description of events into a description of the premises—the rules and related choices—driving the player's actions. This strategy, which game theory remarkably assumes every player possesses before the game, is analogous to a program—a list of instructions to be followed based on various conditions. A player's strategy is not a summary of the rules of the game, but rather a list of choices to be followed—it is, to return to a distinction introduced in chapter 1, a product of "programming" rather than coding. Or, to put it slightly differently, understanding game strategy as a program highlights the fact that a program does not simply establish a universe as Weizenbaum argues; it is one possible strategy devised within an overarching structure of rules (a programming language). A strategy/program thus emphasizes the programming/economic agent as freely choosing between choices.[108]

This program/strategy has been the basis of much of the criticism directed against game theory, such as Gregory Bateson's contention:

> What applications of the theory of games do is to reinforce the player's acceptance of the rules and competitive premises, and therefore make it more and more difficult for the players to conceive that there might be other ways of meeting and dealing with each other. . . . Von Neumann's "players" differ profoundly from people and mammals in that those robots totally

> lack humor and are totally unable to "play" (in the sense in which the word is applied to kittens and puppies).[109]

Bateson is absolutely correct in his assessment: in outlining such a comprehensive version of a strategy, game theory assumes a player who could only be—or later would become—an automaton. Furthermore, von Neumann admits that game theory is prescriptive rather than descriptive. He writes, "the immediate concept of a solution is plausibly a set of rules for each participant which will tell him how to behave in every situation which may conceivably arise."[110] Thus, game theory presumes a strategy and the production of a strategy, as well as the replacement of a detailed description of every action with a more general procedural one. A strategy is something an automaton—or more properly a programmer—working non-"interactively" with a computer has. Game theory's assumptions again resonate with those of neoliberalism (Milton Friedman, to take one example, theorizes the day-to-day activities of people as analogous to those of "the participants in a game when they are playing it").[111]

Words, as instructions that stand in for deeds, are also crucial to von Neumann's desire to make his machines "universal." Von Neumann approaches the concept of universality through an interpretation of Alan Turing's "On Computable Numbers, with an Application to the Entscheidungsproblem," the 1936 paper that initially inspired McCulloch and Pitts.[112] In this paper, Turing shows that Hilbert's *Entscheidungsproblem* (the decision problem) cannot have a solution through theoretical machines, analogous to a "man," that can compute any number. He also posits the existence of a "universal machine," "a single machine which can be used to compute any computable sequence."[113] Von Neumann, in a rather historically dubious move, equates abstract or universal Turing machines with higher-level languages.

To make this argument, von Neumann separates codes into two types: complete and short. In computing machines, complete codes "are sets of orders, given with all necessary specifications. If the machine is to solve a specific problem by calculation, it will have to be controlled by a complete code in this sense. The use of a modern computing machine is based on the user's ability to develop and formulate the necessary complete codes for any given problem that the machine is supposed to solve."[114] Short codes, in contrast, are based on Turing's work, in particular his insight that "it is possible to develop code instruction systems for a computing machine which cause it to behave as if it were another, specified, computing machine."[115] Importantly, Turing himself did not refer to short or complete codes, but rather to instructions and tables to be mechanically—meaning faithfully—followed. Despite this, von Neumann argues that a code following Turing's schema must do the following:

> It must contain, in terms that the machine will understand (and purposively obey), instructions (further detailed parts of the code) that will cause the machine to examine every order it gets and determine whether this order has the structure appropriate to an order of the second

machine. It must then contain, in terms of the order system of the first machine, sufficient orders to make the machine cause the actions to be taken that the second machine would have taken under the influence of the order in question.

The important result of Turing's is that in this way the first machine can be caused to imitate the behavior of *any* other machine.[116]

Thus, in a remarkably circular route, von Neumann establishes the possibilities of source code as logos: as something iterable and universal. Word becomes action becomes word becomes the alpha and omega of computation.

Enduring Ephemeral

Crucially, memory is not a static but rather an active process. A memory must be held in order to keep it from moving or fading. Again, memory does not equal storage. Although one can conceivably store a memory, *storage* usually refers to something material or substantial, as well as to its physical location: a store is both what is stored and where it is stored. According to the OED, to store is to furnish, to build stock. Storage or stocks always look toward the future. In computer speak, one reverses common language, since one stores something in memory. This odd reversal and the conflation of memory and storage gloss over the impermanence and volatility of computer memory. Without this volatility, however, there would be no memory. To repeat, memory stems from the same Sanskrit root for *martyr*. Memory is an act of commemoration—a process of recollecting or remembering.

This commemoration, of course, entails both the permanent and the ephemeral. Memory is not separate from questions of representation or enduring traces. Memory, especially artificial memory, traditionally has brought together the permanent and the ephemeral; for instance, the wax tablet with erasable letters (the inspiration for classical mnemotechnics). As Frances A. Yates explains, the rhetorician treated architecture as a writing substrate onto which images, correlating to objects to be remembered, were inscribed. Summarizing the *Rhetorica Ad Herennium*, the classic Latin text on rhetoric, she states:

The artificial memory is established from places and images . . . the stock definition to be forever repeated down the ages. A *locus* is a place easily grasped by the memory, such as a house, an intercolumnar space, a corner, an arch, or the like. Images are forms, marks or simulacra . . . of what we wish to remember. For instance, if we wish to recall the genus of a horse, of a lion, of an eagle, we must place their images on a definite *loci*.

The art of memory is like an inner writing. Those who know the letters of the alphabet can write down what is dictated to them and read out what they have written. Likewise those who have learned mnemonics can set in places what they have heard and deliver it from memory. "For the places are very much like wax tablets or papyrus, the images like the letters, the arrangement and disposition of the images like the script, and the delivery is like the reading."[117]

Visiting these memorized places, one revives the fact to be recalled. This discussion of memory offers a different interpretation of the parallels between human and computer memory. The rhetorician was to recall a physical space within her mind—the image is not simply what is projected upon a physical space, but also the space for projection. Similarly, computer memory (which, too, is organized spatially) is a storage medium *like* but not quite paper. Both degenerate, revealing the limitations of the simile.

Memory as active process is seen quite concretely in early forms of "regenerative memory," from the mercury delay line to the Williams tube, the primary memory mentioned earlier. The serial mercury delay line (figure 4.9) took a series of electrical pulses and used a crystal to transform them into sound waves, which would make their way relatively slowly down the mercury tube. At the far end, the sound waves would be amplified and reshaped.[118] One tube could usually store about a thousand binary bits at any given moment.

Another early memory device, the Williams tube (figure 4.10), derived from developments in cathode ray tubes (CRTs); the television set is not just a computer screen, but was also once its memory. The Williams tube takes advantage of the fact that a beam of electrons hitting the phosphor surface of a CRT not only produces a spot of light, but also a charge. This charge will persist for about 0.2 seconds before it leaks

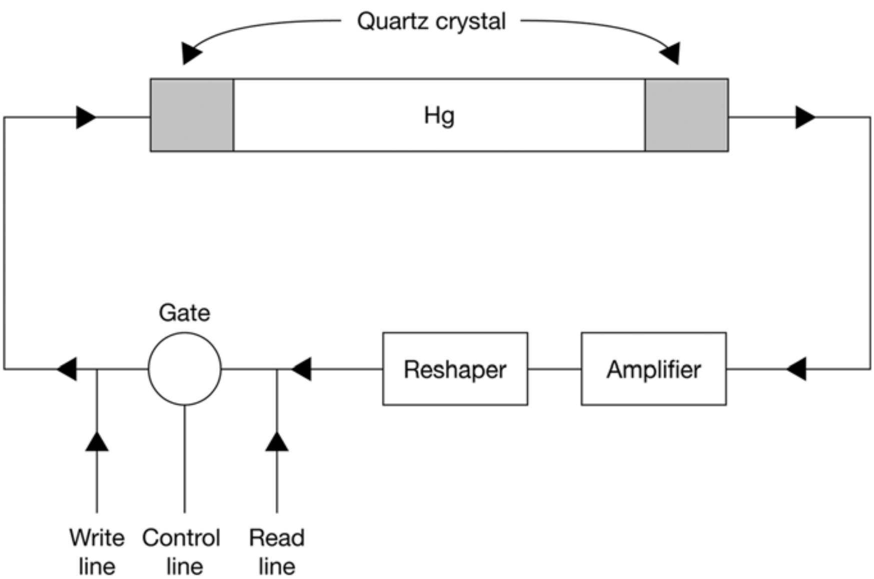


Figure 4.9
Schematic of the mercury delay line

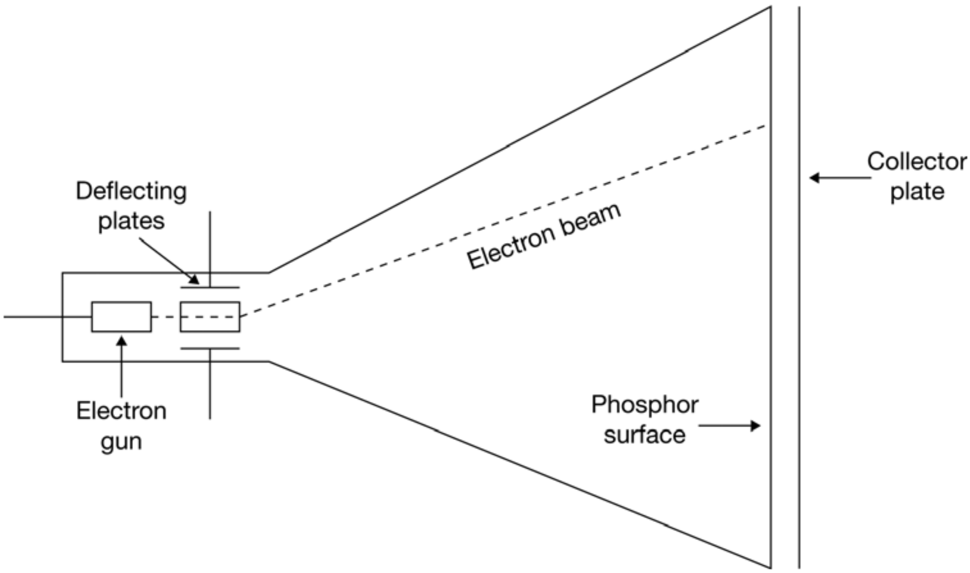


Figure 4.10
Schematic of the Williams tube

away and can be detected by a parallel collector plate. Thus, if this charged spot can be regenerated at least five times per second, memory can be produced in the same manner as the mercury delay tube. Current forms of computer memory also require regeneration.

Today's RAM is mostly volatile and based on flip-flop circuits and transistors and capacitors, which require a steady electrical current. Although we do have forms of nonvolatile memory, such as flash memory, made possible by better-insulated capacitors, they have a limited read-write cycle. Memory traces, to repeat Derrida's formulation, "produce the space of their inscription only by acceding to the period of their erasure."[119]

Thus, as Wolfgang Ernst has argued, digital media is truly *time-based media*, which, given a screen's refresh cycle and the dynamic flow of information in cyberspace, turns images, sounds, and text into a discrete moment in time. These images are frozen for human eyes only.[120] Information is dynamic, however, not only because it must move in space on the screen, but also, and more important, because it must move within the computer and because degeneration traditionally has made memory possible while simultaneously threatening it. Digital media, allegedly more permanent and durable than other media (film stock, paper, etc.), depends on a degeneration so actively denied and repressed. This degeneration, which engineers would like to divide into useful versus harmful (erasability versus signal decomposition, information versus noise), belies and buttresses the promise of digital computers as permanent memory machines. If our machines' memories are more permanent, if they enable a permanence that we seem to lack, it is because they are constantly refreshed—rewritten—so that their ephemerality endures, so that they may "store" the programs that seem to drive them. To be clear, this is not to say that information is fundamentally immaterial; as Matthew Kirschenbaum has shown in his insightful *Mechanisms: New Media and the Forensic Imagination*, information (stored to a hard drive) leaves a trace that can be forensically reconstructed, or again, as I've argued elsewhere, for a computer, to read is to write elsewhere.[121] This is to say that if memory is to approximate something so long lasting as storage, it can do so only through constant repetition, a repetition that, as Jacques Derrida notes, is indissociable from destruction (or in Bush's terminology, forgetting).[122]

This enduring ephemeral—a battle of diligence between the passing and the repetitive—also characterizes content today. Internet content may be available 24/7, but 24/7 on what day? Further, if things constantly disappear, they also reappear, often to the chagrin of those trying to erase data. When A3G (article III groupie), the gossipy conservative and supposedly female author of underneaththeirrobes.blogs.com—a blog devoted to Supreme Court personalities—came out as a thirty-year-old Newark-based U.S. attorney named David Lat in an interview with the *New Yorker*, his site was temporarily taken down by the U.S. government.[123] Archives of his site—and of every other site that does not reject robots—however, are available at the Internet Wayback Machine (IWM, web.archive.org) with a six-month delay.

Like search engines, the Internet Wayback Machine comprises a slew of robots and servers that automatically and diligently, and in human terms, obsessively, back up most web pages. Also like search engines, they collapse the difference between the Internet, whose breadth is unknowable, and their backups; however, unlike search engines, the IWM does not use the data it collects to render the Internet into a library, but rather use these backups to create what the creators of the IWM call a "library of the Internet." The library the IWM creates, though, certainly is odd, for it has no coherent shelving system: the IWM librarians do not offer a card catalog or a comprehensive, content-based index.[124] This is because the IWM's head librarian is a machine, only capable of accumulating differing texts. That is, its automatic power of discrimination only detects updates within a text. The IWM's greatest oddity, however, stems from its recursive nature: the IWM diligently archives itself, including its archives, within its archive.

The imperfect archives of the IWM are considered crucial to the ongoing relevance of libraries. The IWM's creators state: "Libraries exist to preserve society's cultural artifacts and to provide access to them. If libraries are to continue to foster education and scholarship in this era of digital technology, it's essential for them to extend those functions into the digital world."[125] The need for cultural memory drives the IWM and libraries more generally. Noting the loss of early film archives due to the recycling of early film stock, the archivists describe the imperative of building an "internet library":

> Without cultural artifacts, civilization has no memory and no mechanism to learn from its successes and failures. And paradoxically, with the explosion of the Internet, we live in what Danny Hillis has referred to as our "digital dark age."
>
> The Internet Archive is thus working to prevent the Internet—a new medium with major historical significance—and other "born-digital" materials from disappearing into the past. Collaborating with institutions including the Library of Congress and the Smithsonian, we are working to preserve a record for generations to come.[126]

The IWM is necessary because the Internet, which is in so many ways *about* memory, has, as Ernst argues, no memory—at least not without the intervention of something like the IWM.[127] Other media do not have a memory, but they do age and their degeneration is not linked to their regeneration. As well, this crisis is brought about because of this blinding belief in digital media as cultural memory. This belief, paradoxically, threatens to spread this lack of memory everywhere and plunge us negatively into a way-wayback machine: the so-called "digital dark age." The IWM thus fixes the Internet by offering us a "machine" that lets us control our movement between past and future by regenerating the Internet at a grand scale. The Internet Wayback Machine is appropriate in more ways than one: because web pages link to, rather than embed, images, which can be located anywhere, and because link locations always change, the IWM preserves only a skeleton of a page, filled with broken—rendered—links and images (figure 4.11). The IWM, that is, only backs up certain data types. These "saved"

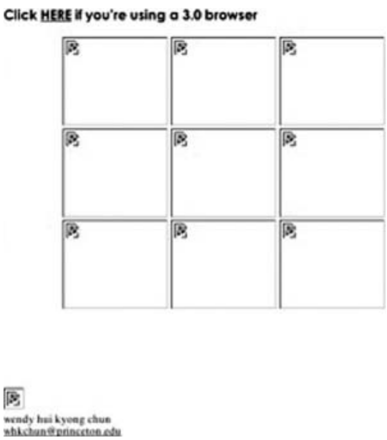


Figure 4.11
Screenshot of IWM backup of <http://www.princeton.edu/~whkchun/index.html>

pages are not quite dead, but not quite alive either, for their proper commemoration requires greater effort. These gaps not only visualize the fact that our constant regenerations affect what is regenerated, but also the fact that these gaps—the irreversibility of this causal programmable logic—are what open the World Wide Web as archive to a future that is not simply stored upgrades of the past.

Repetition and regeneration open the future by creating a nonsimultaneous new that confounds the chronological time these processes also enable. Consider, for instance, the temporality of weblogs (also known as *blogs*). Blogs seem to follow the timing of newspapers in their plodding chronology, but blogs contain within themselves archives of their posts, making the blog, if anything, like the epistolary novel. Unlike the epistolary novel, which, however banal, was focused on a plot or a moral, the blog entries are tied together solely by the presence of the so-called author. What makes a blog "uninteresting" is not necessarily its content, which often reads like a laundry list of things done or to do, but rather its immobility. The ever-updating, inhumanly clocked time in which our machines and memories are embedded and constantly refreshed makes the blog's material stale. The chronology, seemingly enabled by this time, is also compromised by these archives and the uncertainty of their regular reception. An older post can always be "discovered" as new; a new post is already old. This nonsimultaneousness of the new, this layering of chronologies, means that the gap between illocutionary and perlocutionary in high-speed telecommunications may be dwindling, but—because everything is endlessly repeated—response is demanded over and over again. The new is sustained by this constant demand to respond to what we do not yet know, by the goal of new media czars to continually create desire for what one has not yet experienced.

Digital media networks are not based on the regular obsolescence or disposability of information, but rather on the resuscibility or the undead of information. Even text messaging, which seems to be about the synchronous or the now, enables the endless circulation of forwarded messages, which are both new and old. Reliability is linked to deletion: a database is considered to be unreliable (to contain "dirty data"), if it does not adequately get rid of older, inaccurate information. Also, this repetition, rather than detracting from the message, often attests to its importance. Repetition becomes a way to measure scale in an almost inconceivably vast communications network.

Rather than getting caught up in speed then, what we must analyze, as we try to grasp a present that is always degenerating, are the ways in which ephemerality is made to endure. Paul Virilio's constant insistence on speed as distorting space-time and on real-time as rendering us susceptible to the dictatorship of speed has generated much good work in the field, but it can blind us to the ways in which images do not simply assault us at the speed of light.[128] Just because images flash up all of a sudden does not mean that response or responsibility is impossible, or that scholarly analysis is no longer relevant. As the news obsession with repetition reveals, an image does not flash up only once. The pressing questions are: why and how is it that the ephemeral endures? And what does the constant repetition and regeneration of information effect? What loops and what instabilities does the enduring ephemeral introduce to the logic of programmability? What is surprising is not that digital media fades, but rather that it stays at all and that we remain transfixed at our screens as its ephemerality endures.

Conclusion: In Medias Res

> No matter how forewarned we are, thanks to the forearmaments of the knowledge of the secret of commodity exchange and its resulting fetishism, as long as exchange (language) goes on we are powerless to overcome its difficulties. And knowing makes it more scary. "Je sais bien, mais quand même." As Marx says, this is the path of madness: "If I state that coats or boots stand in a relation to linen because the former is the universal embodiment of abstract human labor, the craziness . . . of the expression hits you in the eye. But when the producers of coats and boots bring these commodities into relation with linen . . . the relation . . . appears to them in this crazy . . . form.". . . "Humanity" is this madness, its subject and its object. It is not simply the ignorance of not knowing what to do; it is rather the terror of still having to do, without knowing. And we have no magic caps, only ghosts and monsters.
>
> —Thomas Keenan[1]

This book has traced the emergence of programmability through various theoretical and historical threads: code—both computer and genetic—as logos, user as sovereign, interfaces as "enlightening" maps, computer as metaphor for metaphor, and programmability as both thriving on and annihilating memory. It explores the extent to which computers, understood as networked software and hardware machines, are—or perhaps more precisely set the grounds for—neoliberal governmental technologies. And it examines how computers accomplish this not simply through the problems (population genetics, bioinformatics, nuclear weapons, state welfare, and climate) they make it possible to both pose and solve, but also through their very logos, their embodiment of logic.

The book began in part I with the question of code as logos, that is, with a "sourcery" that posited code written in higher-level programming languages as automatically and unfailingly "doing what it says." As the perfect performative utterance, code brought together two separate powers, the legislative and the executive, making execution and hardware largely irrelevant. This sourcery also opened part II, which posited genetics and computer code as complementary strands of a stylized double helix. Notably, code as logo*s* within genetics precedes (rather than simply follows) its