# My Fascination with Computing History

John Impagliazzo

`John.Impagliazzo@hofstra.edu`

**Abstract.** This narrative on computing history reflects the experiences of the author and his involvement with computing history over a quarter century. The discussion portrays a transition from loathing history as a student to embracing computing history as a professional. The author shows how storytelling can produce interesting excursions on technical subjects and ways in which teaching computing with history can elevate student interest. He also provides examples showing ways in which historical events could complement computing studies. The article also explains how the author's earlier efforts in using history to teach computing led to a landmark publication and subsequent activities within IFIP leading to conferences and related publications.

**Keywords:** Computing history, computing education, history and computing, human factors in computing.

## 1    Background

In the course of one's life, certain events transcend their primal characterizations. As a student, I had always found history uninteresting. For me, the subject was one of memorization of certain facts such as persons, dates, and places. The usual basis for class and course examinations was memorization and little critical thinking was ever involved. That pattern continued to the point where I would only study history when required to do so. With a few exceptions in my circle of student colleagues, almost everyone knew I disliked history.

As years passed, I began to kindle a new liking for the subject. Pioneers in mathematics, engineering, physics, and computing began to fascinate me. People such as Galileo, Newton, Euler, and von Neumann began to influence my thinking with the realization that contributions to their subjects were more than an equation or a process. These were people, humans, who had advanced their subject areas to new heights. Should I, we, not know more about them?

The juncture of this epiphany occurred in the 1980s. I had become a close friend of the famous mathematician, Marshall H. Stone. I had begun to write his mathematical biography and in the many meetings I had with Marshall, he would always bring up "Johnny", referring to eminent mathematician and physicist John von Neumann. Although von Neumann died in 1957, a worldwide celebration of his achievements was lacking. I can still hear Marshall say, "We must do something for Johnny".

As it happened, I eventually organized the conference called the "Legacy of John von Neumann" in May of 1988 at Hofstra University. Famous scientists, engineers, mathematicians, and Nobel laureates who knew of or had an affinity to von Neumann

and his works spoke at the conference. Even von Neumann's family members and some of his students participated in the event. For me, this gathering of scientific dignitaries became the turning point of my new fascination with history and in particular, computing history. The proceedings from this event [2], which I co-edited, still receive international acclaim.

In the late 1980s, I began incorporating elements of history in the computing courses I taught. Almost every class included a morsel or story on some historical nuance related to the topic at hand. If we were discussing the von Neumann bottleneck, I would ask, "Who was John von Neumann"? If we were discussing a finite automaton, I would ask, "Who developed graph theory"? Of course, few if any students knew the answers to these questions. They would then do a lookup somehow (remember, web browsers did not exist then), and the next day I would have a brief discussion surrounding the findings from the students. The most embarrassing question for students was,

> *Who invented the computer?*

After dismissing responses such as Bill Gates or Steve Jobs, the discussions often became quite interesting – even heated or bizarre. I would berate the class by saying something like,

> *What a disgrace! Computer science majors and they don't even know who invented the computer.*

with the hope of garnering their greater attention and perk their curiosity.

In the mid-1990s, ACM designated me to chair the 1996 ACM Special Interest Group on Computer Science Education (SIGCSE) conference in Philadelphia, Pennsylvania. This 1996 ACM gathering included more than the SIGCSE event. It also included the beginning of the fiftieth anniversary celebration of ACM, the fiftieth anniversary of the ENIAC computer with its ceremonial re-launching at the University of Pennsylvania, the last ACM Computer Science Conference, and the chess match between the IBM Deep Blue computer and the world chess master and champion Garry Kasparov. The year before, John A.N. Lee from Virginia Tech published his book on computing pioneers [11]; I invited JAN to be one of the keynote speakers at the SIGCSE event. JAN was the former editor-in-chief of the *IEEE Annals of the History of Computing*; he was also the chair of Working Group 9.7 (WG 9.7) on the History of Computing of the International Federation for Information Processing (IFIP) [3]. JAN and I soon became friends, a friendship that furthered my growing interest in computing history.

## 2    Storytelling

As already mentioned, I had been using history in computing classes since the late 1980s; storytelling was one of my approaches. The study of computing hardware and software could be a "dry" experience for students; teachers usually present technical material in sterile ways. By using the technique of storytelling, I provided background to the theories of hardware and software topics. In this manner, students would gain a better understanding of the topic under development and learn theoretical concepts in

the context of social, political, or economic conditions. I have found that students often gain a better understanding of the subject when they learn how diverse events and ideas could fuse together to stimulate discussion and inventive ideas on the evolution of new and modern devices and machines. Experience shows that when I involve history and storytelling in computing courses, they make the learning experience much more interesting and meaningful.

Introductory courses, for example, often take an "overview" approach. This allows flexibility in the way a teacher could present topics. Students attending introductory courses often come from a variety of specialties; this heterogeneous mix of students creates a perfect environment to use historical morsels to elevate student interest; it also allows instructors to explore and use inventive measures to retain student attention. Computing history is one way to merge technical and non-technical topics to foster a constructive learning experience.

## 3      Teaching Computing with History

George Santayana (1863-1952) had alerted us on the importance of history. One variation of his saying is, "Those who cannot learn from the past are condemned to repeat it" [12]. As teachers, we should not ignore the past and the lessons learned for a more promising future. We could use history to resurrect ideas and concepts that either progress had overtaken or were ahead of their time. Without computing history, the core of computing topics becomes only inanimate objects of hardware and software without human connection. It is a teacher's duty to integrate the technical with the non-technical and to connect events to form a tapestry of knowledge for students. Hence, the use of history in computing helps students and professionals make sound technical and business decisions based on their knowledge and experience of the past. Therefore, we see that learning computing topics in context can greatly enhance the way we learn technical aspects of the subject. Learning topics in isolation could be meaningless.

One of the downfalls of the computing profession is the poor way it promotes its own heroes. Unlike other areas such as mathematics, engineering, and science, teachers of computing tend to shy away from hailing the achievements of others. As a result, students hardly hear of the heroes from their profession, perhaps with the exception of Bill Gates and Steve Jobs. How sad. Our computing field was the invention of pioneers. We should not forget them or deny them their due credit. They have paved the way for this relatively new and emerging field. Indeed, they are our heroes.

Students entering their first year of college in the fall of 2012 were likely born in 1994, shortly after the invention of web browsers that we take for granted today. Do they know the story of the trailblazers of this use of the internet? Do they know how this computing network has reached the level of achievement we inherited and use today? I doubt it. Hence, exploring the achievements and challenges of our internet heroes would be one way to expand students' knowledge and engage them in a subject having both technical and non-technical ramifications. An exposure to computing history in this regard would reveal many nuances that could broaden students' thinking and they would likely become more inquisitive and perhaps even more inventive.

Students need history because all mature disciplines transmit their origins and stories to new advocates. It is an acculturation process and teachers usually find it difficult to visualize items of the present time. So, how can we understand them? Older computing systems were simpler and easier to understand. Hence, students need to understand that change is the only constant in the computing field. That is, all elements of computing are evolving and improving. Indeed, students deserve to know the truth regarding the evolution of the subject they are learning.

# 4     Examples of Using History

When people used the word 'computer' in the early days, they generally meant a mechanical, electro-mechanical, or an electronic device to do calculations. In fact, until the mid-1940s, a computer was actually a *person* who did calculations. A more formal and more modern definition of a computer is an electronic calculating device that contains a memory component. Examples of today's computers include desktop and personal computers, mobile phones, smartphones, tablet devices, and a myriad of other electronic devices used to manipulate and store data as well as communicating information.

Let us consider a few cases to show how one could use history to enhance computing topics. Here, we focus on topics that one might teach and show how the inclusion of individuals and events can enrich the topic at hand. We also note that we should not classify any of the machines mentioned in this section as computers in the modern sense. The reason is that the modern definition of a computer involves an active memory component that can store operating instructions and data. These machines are void of that component.

## 4.1     Early Mechanical Computers

Perhaps the earliest (about 3000 BCE) mechanical device for doing arithmetic calculations is the abacus, a device still in use today in some parts of the world. More sophisticated though less practical devices for doing simple arithmetic are the gear-driven adding machine called the "Pascalene" of Blaise Pascal (1642) and a calculator using a stepped cylinder gear called the "Stepped Reckoner" of Gottfried Leibniz (1674). Although these mechanical calculators do not fall under the category of "computer", students could explore how they provided some of the ideas and building blocks of the computing machines of today.

In 1822, Charles Babbage began to design and build the Difference Engine, a mechanical machine intended to calculate tables. Babbage never completed the gear-driven machine, but it was a precursor to his follow-up invention of the Analytical Engine in 1832. Although never completed, this mechanical engine became the basis for engineering modern computing machines. For this reason, some historians consider Babbage to be one of the inventors of the computer. Allowing students to explore this acclamation provides a sound entrée to other interesting overtones related to Babbage and computers.

One hundred years had lapsed without any real progress in developing computing machines. Enter Konrad Zuse. A quick search of the person would reveal that he indeed produced original computers in the late 1930s and early 1940s. Some historians consider Zuse as one of the inventors of a computer. His machines were electro-mechanical devices. A topic such as this would be very appropriate in the early studies of computing. Zuse's machines (called Z1, Z2, and Z3 in their early developments) used physical relays (devices that close and open) to do binary switching. Zuse also invented the first high-level programming language, Plankalkül, to play games on the machines he invented. I wonder how many students or teachers know this fact. A good way to engage students is to have them research the works of Zuse and report their findings.

John Atanasoff designed the ABC computer in 1937 and built it with his graduate student Clifford Berry at Iowa State University. The ABC was an all-electronic (no mechanical part) machine. This special-purpose computer was able do only one thing—it solved systems linear equations. The computer became operational in 1942. The machine implemented the use of binary digits, was totally electronic, was able to systemize memory and computation (similar to a primitive operating system), and used regenerative capacitor memory, items found in every modern computer. Many historians regard Atanasoff as the (forgotten) inventor of the computer. Students and teachers who research the work of Atanasoff will find many interesting revelations.

At the Moore School of the University of Pennsylvania in the early 1940s, John Mauchly and his graduate student and engineer Presper Eckert were building a machine to do military and weather calculations. Called the Electronic Numerical Integrator and Computer (ENIAC) and funded by the federal government, historians consider this machine as the first general-purpose computer because one could program it by changing its hardware circuit boards. The ENIAC was a colossal machine; it contained over 18,000 vacuum tubes (valves) and it had a high probability for system failure. Students who research the story of this machine and the people involved with it should find the investigation quite informative.

One should note that the ABC and the ENIAC machines did not exist without controversy. Eckert and Mauchly patented the ENIAC in the late 1940s. However, many believed that the patent was not valid because the patent office did not consider the "prior art" requirement for patents, especially since Mauchly visited Atanasoff in 1941 and learned about the features of the ABC computer, some of which were incorporated in the ENIAC. After a long legal battle, an adjudication revealed that Eckert and Mauchly did not themselves first invent the ENIAC; instead they derived the subject matter from Atanasoff. Students should be fascinated from the ramifications surrounding these two machines.

## 4.2    Universal Computers

It is worth mentioning that not all computers need to be physical machines. Since 2012-2013 represents the centennial of the birth of Alan Turing, let us discuss briefly the idea of his universal computer. A universal machine (computer) is one that could calculate any computable function. We will skip the details of this, only to mention

that Turing proved that every computable function is solvable by such a machine, which we call a Turing Machine. This topic and all its ramifications appear in courses on computability, theory of computing, and other related subjects. So, how does one enrich this topic using history?

Some historians consider Alan Turing as one of the original inventors of the computer. An interesting exercise here is to have students explore the veracity of this statement and have them investigate other inventors as mentioned above. A study of the 1937 universal machine would reveal that this machine existed only in theory; that is, it was not physical entity. The immediate debate that would occur would involve the challenge of whether a non-physical machine would qualify as a "computer", which could form the basis for a philosophical dialogue and debate among students. No one has actually built this theoretical machine; however, many simulations of it exist. In fact, in my classes I would often ask students to build a simulator for such a machine as a group project or as an individual exercise. The opportunity to program an original computer stimulates great interest among students, given that they have at their disposal modern languages and tools to simulate a working universal machine. Indeed, one could hold a class contest to see which student or group makes the best simulator.

The development of universal simulators is of great interest to students. The simulators allow them to examine this machine and modify its parameters. It helps to show that Turing's claim for universality is important to the understanding of the foundations of the computing field. It also provides the occasion to discuss the life story of Alan Turing and his other contributions to the field of computing such as being a pioneer in computational biology, his code breaking activities at Bletchley Park during World War II, and his pioneering work in machine learning. The study of theoretical computing could be a dull topic for many students. However, with the infusion of computing history, this topic could easily burst into new episodes to enrich the mind and elevate the spirit of most students.

## 4.3    Computer Memory and Pipelining

Computer memory components can take various forms. Electronic single memory components came into existence in 1947 when F.C. (Freddie) Williams and Tom Kilburn developed a cathode-ray tube memory device and in 1948 when Andrew D. Booth developed the magnetic memory drum. These single memory devices soon reverted to electronic circuit storage devices in the 1950s. Teachers of computing could use these facts as a launch point for further discussion on the development of computer memory and the concept of hierarchical memory. For example, what was the significance of the Atlas computer at Manchester University? What did Maurice Wilkes propose in a 1965 short paper on caching? How did IBM commercially implement the first cache machine in the IBM 360/85 machine? All of this has led to great efficiency advances with memory hierarchies and virtual memory schemes considered common in modern day machines. Asking students to explore and research these inventions engages them in their studies and expands their understanding of the topics they are learning.

Another efficiency scheme is pipelining, an architectural scheme that enables a computer to do multiple activities within a given timeframe. The Control Data CDC 6600 from 1964, often considered "the first supercomputer", was the first commercial machine that used pipeline methodologies and dynamic scheduling. This strategy evolved to the use of multiple function units, score boarding, and new algorithms such as the Tomasulo algorithm. In turn, these novel processes evolved to reduced instruction set computers (RISC) in the mid-1980s through the contributions of John Cocke, which led to the series of IBM Power machines. Pipelining and its extensions are commonplace in today's computers. Asking students to explore these aspects of computing should perk and increase their interest regarding the manner in which companies build computers today.

### 4.4    Reflections

With respect with Sections 4.1 and 4.2 above, we should note that John von Neumann first proposed a blueprint of a modern computer in his 1945 white paper [13], from which we attribute the idea of the "von Neumann architecture" in his honor. Computers in the modern sense first appeared in the late 1940s. Before that time, computers were electronic computational devices such as the ABC special-purpose computer of Atanasoff in 1942 and the ENIAC general-purpose computer of Eckert and Mauchly in 1946. The ability of a machine to store dynamically and process data forms the basis of a modern meaning of a computer. Precursors to the computer in its modern meaning do provide much fodder for student-teacher discussion and interaction. Broadening the non-technical dimensions and understanding of the computing field is a recipe for sound and informed computing practices in one's career.

## 5    Engaging Students

Students do find the development and episodes of the computers mentioned in Section 4 intriguing. So, how does a teacher incorporate some of this learning in an introductory course? One presumption is that instructors have some familiarity with the work of pioneers or they can quickly learn this aspect of computing. It is good to do some show-and-tell activities of relays and valves (or vacuum tubes) if available or at least show pictures of them. In this modern day of smart phones and tablets, students are beginning to appreciate the challenges of the past; they also learn how past challenges have led to the devices they commonly use today.

We could ask some interesting questions related to computer hardware elements and devices. For example, have students ever held, touched, or even seen the internals of a computer with its components?

Computing topics often center on software and applications. Few students receive exposure to the electrical and electronic elements of a computer and even fewer have touched resistors, capacitors, transistors, vacuum tubes, memory chips, floppy disks, or motherboards. Generations of computer developments offer students the opportunity to

examine these items and try to understand the challenges of space and heat problems others confronted with early machines. Hence, a discussion of people such as Zuse, Turing, or Atanasoff opens doors for exciting discussions, particularly the challenges they faced in building original computers. The discussion injects "humanizing" elements in the teaching of computing and technology.

Another challenge involves the teachers themselves. Are computing teachers and instructors sufficiently conversant with the history of the subject they teach? Probably not. It is unlikely that teachers have formally studied computing history; the knowledge they have is self-taught. Hence, teachers of computing and its technologies often shy away from its history in favor of technical approaches, even though they realize that knowledge of history could be useful in avoiding future pitfalls. A brief discussion of computing pioneers among students could greatly enrich computing topics and allow students to explore new avenues of learning.

## 6    The "IFIP Report"

At the IFIP World Computer Congress (WCC) in Canberra, Australia, JAN Lee and I met and discussed the development of a document that could become a guide for teachers to embed history in the computing curriculum. Of course, this was music to my ears since I was already doing those activities for almost a decade. JAN and I strategized ways to accomplish this end, particularly if it had international appeal. Since we were both at WCC, it became immediately clear that IFIP could be that vehicle. Since JAN already represented the Working Group 9.7, a working group of IFIP's Technical Committee 9 (TC9), it was useful to include a representative from Technical Committee 3 (TC3) on Education. Gordon Davies, then chair of computer science at the Open University in the U.K. and active in TC3, became a logical choice. We also recruited two computing historians, Michael Williams of Calgary and Martin Campbell-Kelly of Warwick, to form a team of five. The group designated me as their leader.

Over a period of two years, the team developed drafts of what would become the content of this document; it would eventually receive the approval of IFIP. One contrasting and diverse opinion was whether history in computing should be taught solely as a course or sprinkled throughout the curriculum. Another point of contention was whether one should teach a history course as a chronology of events or in some other manner. For the former point, the team decided to include both opinions to allow flexibility in approach. For the second point, the document suggested a "whatever works best" approach for a given situation. The team did not want to be prescriptive in its recommendations to IFIP; it simply wanted to increase the awareness of computing teachers allowing them to develop their own pathways on ways to use computing history. It also wanted to make teachers aware of resources available to them to enrich a student's learning experience through the inclusion of historical insights and dimensions.

The team titled the final document "History in the Computing Curriculum" and presented it to the IFIP constituents (TC9 and TC3) for approval in 1998, which it received. The document, also known as the "IFIP Report", was subsequently published in the *IEEE Annals of the History of Computing* in January of 1999 [4]. The

contents of the thirteen-page report included sections such as (a) need for history content, (b) curriculum content, (c) implementation of a basic curriculum, and (d) assessment. Of significant interest was the Appendix to the report. It included among other things a chronology of historical events in computing, relevant websites, audio-visual materials, some examples of learning clusters, references to existing computing history courses, sample course syllabi, and listings of journals, textbooks, and other works on computing history. In all, the report received wide acclaim and it is still referenced to this day.

## 7    Experiences as WG 9.7 Chair

It would be good to explore how different circumstances have led me to acquire a zeal for organizing conferences and publishing books on computing history. The principal force for me was and still is promoting the use of history to motivate student learning. As mentioned earlier, my use of computing history was a normal theme in my courses since the late 1980s. I was and am passionate about computing history because students and professionals begin to think "outside the box" and look beyond machines and languages to see how the history of a topic broadens a person's outlook on the world. Computing history also allows one to investigate and research topics in the context of lifelong learning and make connections between technology and human reality. Indeed, computing history enhances the study of computing and expands student learning of the subject.

The von Neumann conference that I chaired in 1988, coupled with my desire to include computing history in my courses, coupled with the publication of the "IFIP Report" have all contributed to my escalation and interest in computing history. I joined WG 9.7 and soon thereafter started attending TC9 meetings. In 2001, JAN Lee decided to step down as chair of WG 9.7 and the TC9 appointed me as his successor. At that time, the working group was very small with only about a half-dozen members. I decided to change that image of "just a few" and by the end of my six-year tenure, well over sixty people were members of the working group, most of whom were historical dignitaries or pioneers in computing. IFIP allows participation as a working group chair only for two, three-year terms. In 2007, Arthur Tatnall succeeded me as its new chair.

The infusion of computing history through WG 9.7 provided a pathway toward expanding the message of computing history. The organization of events to herald the importance of history was destined to become pivotal benchmarks toward promoting that idea. As I am very fond of Nordic culture, I thought it best to launch an event that would "unite" the Nordic countries (Denmark, Finland, Iceland, Norway, and Sweden) under a common cause of computing history. Within a year, I was soon able to assemble the organizing and program committees composed primarily of Nordic representatives to launch the first WG 9.7 conference called the "History of Nordic Computing" (HiNC) that had taken place in 2003 in Trondheim, Norway [1]. More than seventy-five computing pioneers attended the event. The enthusiasm shown in Trondheim propelled the organization of a second event in Turku, Finland, in 2007 called HiNC2 [8]. The success of this event soon led to the

organization of HiNC3 in Stockholm, Sweden, in 2010 [9]. Currently, plans are underway for a HiNC4 conference in Copenhagen, Denmark, in 2014. The original thought of uniting the Nordic countries under an umbrella of history has definitely become a sustained reality.

As the HiNC conferences began to gain prominence, another thread of WG 9.7 parallel events was taking place. At a 2003 TC3/WG 3.6 (Distance Education) conference in Geelong, Australia, some TC3 members met with JAN Lee and me to discuss the possibility of a WG 9.7 history conference on education, supported by TC3. Shortly thereafter in 2004, the first history of computing and education (HCE) event [5] had taken place in Toulouse, France, as a track within the IFIP World Computer Congress (WCC). As this was the first education event for WG 9.7, all were rather delighted to know that over two dozen people attended the presentations. At the IFIP WCC is Santiago, Chile, in 2006, the next HCE event occurred [6], where the focus was on history and education in the South American region. The success HCE2 became fodder for another history track within the WCC in Milano, Italy, in 2008 [7]. This was the last of the HCE conferences. At that time, participants decided it would be best to decouple the combined education and history themes; future WG 9.7 events would focus on computing history in the broader sense, which would even include education.

Perhaps the most extraordinary event I organized as WG 9.7 chair was the 2006 conference on "Perspectives on Soviet and Russian Computing" (SoRuCom) in Petrozavodsk, Russia [10]. This event had taken about five years in the making and necessitated traveling to Russia (at my own expense) several times. Petrozavodsk is approximately 435 kilometers north of St. Petersburg and 980 kilometers northwest of Moscow. The Petrozavodsk State University had taken an interest in hosting this event, which brought together over a hundred-fifty Soviet scientists, academicians, and interested parties to have a franc exchange of ideas and points of view. One must remember that in the time of the Former Soviet Union, people worked on projects under a shroud of secrecy. It was astonishing to see many of the computing pioneers finally meeting; they were able to discuss their projects openly and without restraint. The SoRuCom conference represented a landmark event in the preservation of the history of Soviet computing. The SoRuCom conference led to the SoRuCom-2 conference held on Novgorod, Russia, in September of 2011; I had little involvement with this conference other than being a member of its organizing committee.

## 8    Summary

We should remember that any course on the fundamentals of computers and computing is by its very nature a course on the history of computing. Unfortunately, teachers usually skip the non-technical nuances and focus only on technical issues. Historical excursions take only a few minutes to execute; the time invested could stimulate student interest and make them more interested in their studies. It is my hope that this journey through time as expounded by this narrative serves as an awakening that will motivate others to add historical dimensions to enrich their courses.

# References

1. Bubenko, J., Impagliazzo, J., Sølvberg, A. (eds.): History of Nordic Computing. Springer, New York (2005)
2. Glimm, J., Impagliazzo, J., Singer, I. (eds.): The Legacy of John von Neumann. Proceedings of Symposia in Pure Mathematics, vol. 50. American Mathematical Society (1990)
3. IFIP, Technical Committee 9, ICT and Society, `http://ifiptc.org/?tc=tc9`
4. Impagliazzo, J., Campbell-Kelly, M., Davies, G., Lee, J.A.N., Williams, M.: History in the Computing Curriculum. IEEE Annals of the History of Computing 21(1), 4–16 (1999)
5. Impagliazzo, J., Lee, J.A.N. (eds.): History of Computing in Education. Kluwer Academic Publishers (2004)
6. Impagliazzo, J. (ed.): History of Computing and Education 2. Springer, Heidelberg (2006)
7. Impagliazzo, J. (ed.): History of Computing and Education 3. Springer, Heidelberg (2008)
8. Impagliazzo, J., Järvi, T., Paju, P. (eds.): History of Nordic Computing 2. Springer, Heidelberg (2009)
9. Impagliazzo, J., Lundin, P., Wangler, B. (eds.): History of Nordic Computing 3. Springer, Heidelberg (2011)
10. Impagliazzo, J., Proydakov, E. (eds.): Perspectives on Soviet and Russian Computing. Springer, Heidelberg (2011)
11. Lee, J.A.N.: International Biographical Dictionary of Computer Pioneers, 816 p. Taylor & Francis (December 1995)
12. McCormick, J.: George Santayana: A Biography. Transaction Publishers, New Brunswick (2003)
13. Von Neumann, J.: First Draft of a Report on the EDVAC. In: Aspray, W., Burks, A. (eds.) Papers of John von Neumann on Computing and Computer Theory. MIT Press, Cambridge (1987)