1991

A Purdue University Course in the History of Computing

Saul Rosen

Report Number:

91-023

A PURDUE UNIVERSITY COURSE IN THE HISTORY OF COMPUTING

Saul Rosen

CSD-TR-91-023
March 1991

A Purdue University Course in the History of Computing

Saul Rosen

CSD-TR-91-023
March 1991

Abstract

University courses in the history of computing are still quite rare. This paper is a discussion of a one-semester course that I have taught during the past few years in the Department of Computer Sciences at Purdue University. The amount of material that could be included in such a course is overwhelming, and the literature in the subject has increased at a great rate in the past decade. A syllabus and list of major references are presented here as they are presented to the students. The course develops a few major themes, several of which are discussed briefly in the paper. The course has been an interesting and stimulating experience for me, and for some of the students who took the course. I do not foresee a rapid expansion in the number of universities that will offer courses in the history of computing. I hope that this paper will give some help and encouragement to others in Computer Science who may wish to undertake to teach such a course.

A Purdue University Course in the History of Computing

Saul Rosen
Department of Computer Sciences
Purdue University
West Lafayette, Indiana.

Introduction

My own interest in the history of computing goes back a very long time. I presented a paper entitled "Programming Systems and Languages - a Historical Survey" at the 1964 Spring Joint Computer Conference. An expanded version of that paper, was included in the book "Programming Systems and Languages", an edited collection of articles that I published in 1967. That book was widely used as a textbook in the early computer science programs in the United States and abroad. Other articles in the book also contained significant amounts of historical information, and students using the book in a course were exposed to some important areas of the history of computing. My article "Electronic Computers-a Historical Survey" was the first article in the new journal "ACM Computing Surveys" in 1969. I felt then, as I have felt ever since, that literacy in the area of computers and computing requires some knowledge of the history of computing.

In the years since the 1960's the courses and textbooks in computer science have become increasingly technical and specialized. Our graduates are often expert in programming in several languages, and they have had courses in computer architecture and data structures and in various technical areas. But most of them know little or nothing about the history of their subject. Lack of historical background in their own area is not unique to computer scientists, but I cannot address this lack in other areas. An attempt to remedy it to some extent for students of computer science has been the motivation for a course in the history of computing that I have developed and taught several times in the Department of Computer Sciences at Purdue University

One might ask whether a course in the history of computing should be taught in a history department or in a computer science department? There are some good arguments for having such a course in a history department, but the history departments at most universities do not have faculty who are qualified to teach such a course, and not many history students are interested in such a course. There are some universities that have History of Science or History of Technology departments, and there have been courses in the History of Computing offered in such departments at the University of Minnesota, at Clemson University, and undoubtedly elsewhere. At the present time I think that in most universities, if such a course is to be offered at all, it would have to be offered in a computer science department. Here at Purdue there are precedents in other departments. I know of a course in the history of mathematics that has been offered occasionally in the Mathematics Dept, and a course in the history of pharmacy in the School of Pharmacy. There must be other such courses in other departments.

Textbooks and Syllabus

In the 1970's when I first thought about offering a course in the history of computing I wondered if I would ever be able to assemble enough material for a full semester course. By the time I was ready to offer the course for the first time in the fall of 1988 I found that I had far more material than I could possibly cover in a single semester. During the 1980's there was a significant increase in the amount of material published in the area of the history of computing. For example, the Annals of the History of Computing and the MIT Press History of Computing series provide a continuing stream of articles and books, many of which are important contributions to the field. It would be interesting to me to offer a full year course or even more in the history of computing, but that is not possible within the framework of the Department of Computer Sciences.

In preparing for most undergraduate courses an instructor can choose from a number of reasonably adequate textbooks. The course is then usually structured in conformity with the selected text. I know of only two books that I would consider for use as textbooks in the history of computing. They are "The Computer from Pascal to von Neumann" by Herman Goldstine, and "A History of Computing Technology" By Michael R. Williams. Goldstine's book was not written as a college textbook, but rather as a book aimed at the scientifically literate public. Goldstine participated in some of the most important projects at the beginning of the age of modern computers. I think that it should be especially interesting for students to read about how it all started, as told by someone who took part in getting it started, and incidentally by someone who writes about it elegantly. As a textbook for my course it is lacking in that it covers only about half of the topics that I would like to include. It contains a good deal of material about analog computing, which is important in its own right, but is an area for which I had planned only minimal coverage.

Williams's book grew out of a course in the history of computing that he has been teaching at the University of Calgary since 1983. A paper describing that course appeared in the Annals of the History of Computing, Vol. 7 No. 3 (1985) p241-244. Williams has done a service to anyone who plans to teach a course in the history of computing by writing a textbook in an area where no textbooks previously existed. However, I did not find his book exactly right for the course I wanted to teach. The book has interesting material about the development of number systems since ancient times, and about early, mostly analog methods of computation. I encourage students to read that material for background information but most of it is outside the scope of my course. I find the book lacking in its discussion of some of the more modern developments in the computer field, and there is no mention of the history of computer software, a topic that I would like to include if only briefly.

Neither of these books is very expensive, especially since Goldstine's book is available in paperback. I therefore find it reasonable to require the students to purchase both. I point out early in the course that much of the course will be based on material not contained in either of the two required texts.

At the first meeting of the course I hand out a fairly long syllabus that is reproduced here in Appendix B. The syllabus is a list of the topics that I would like to cover, but I explain to the class that we cannot possibly cover all of the topics listed, and that the course will consist of a selection from among them. It is clear from this list of topics that this is a course in the history of modern computing. Computing in the ancient world could be a major part of a course in the history of computing but it is not covered in this course. It is also clear that this course is devoted almost exclusively to digital computing. Analog computing was very important in the history of computing, but it appears in the course only in a few topics where it helps clarify the development of digital computing.

References

At the first session I also provide the students with an annotated list of the books that I consider most important for auxiliary reading in the field. All of the books are available in the Purdue Libraries. A short title in parentheses is given for a few entries which I expect to refer to frequently in my lectures. A few items that are not books are included in this list. Appendix C contains the list of references.

Some Major Themes in the History of Computing

It is easy to overwhelm the students with information concerning the succession of people and events that contributed to the history of computing. I try to present the history in terms of a number of important themes, each developed over a sequence of lectures, that contribute to the understanding of the flow of the history. There is not enough time in the one-semester course to handle the important themes adequately. Throughout the course I compromise between an effort to cover a large number of important topics and the need to spend enough time to cover the most important developments adequately.

The General Purpose Programmed Computer

One of the most important themes in the course follows the development of the idea and the implementation of the general purpose programmed digital computer from its beginnings in the nineteenth century through the invention of the stored program electronic computer in the 1940's. Almost everyone agrees that Babbage was the first to come up with the idea of the general purpose programmed computer in the 1830's and 1840's. His own writings about his analytical engine make it clear that he realized almost from the beginning how powerful the concept was. His machine would do more than mathematics, it would even be able to play chess. He was wrong about chess since his machine would not have had enough capacity and would not be fast enough, but he and his friend Ada, the Countess of Lovelace, understood that the analytical engine would be a truly universal machine. There is not nearly enough time to do justice to the story of Babbage and Ada. I find it hard to balance my own admiration for his conception against the brutal fact that he never finished any kind of computing machine.

Alan Turing - From the Theoretical to the Practical

Then there is the very interesting story of Alan Turing. The first time I gave the course I expected the students to know about Turing Machines from some other course, but hardly any of them did. That created a problem, since students of the history of computing should be aware of Turing's famous paper on computability, and have some idea of its significance. The Universal Turing Machine that he describes is a very interesting model of the universal computers that appeared about 10 years later. I like to point out that when he wrote that paper in 1935 Turing was only 23 years old, a beginning graduate student at Cambridge; that he then left England and went to Princeton to work on a Ph.D. with Alonzo Church; that John von Neumann knew Turing at Princeton during that time and so was almost certainly aware of Turing's results when he himself contributed important ideas to the design of the EDVAC.

This discussion of Turing leads naturally to his work on cryptanalysis at Bletchley Park after his return to England in 1939. It is unfortunate that the level of secrecy maintained for many years after World War II, and still maintained in some areas, kept the work at Bletchley Park from getting the recognition that it deserved, and kept it from having the influence on the computer field that it might have had. Our course can cast some light on this and try to clear up some misconceptions. The first Colossus was operational by the end of 1943, and it can claim to be the first working electronic digital computer, but it was not a general purpose programmed computer, and it was not described to the public in any detail until the 1970's. Turing was justly famous for his work on breaking the Enigma cipher, but he had little or nothing to do with Colossus. A number of people who worked at Bletchley Park during the war were involved in post-war computer projects in England. Turing's ACE computer design at the National Physical Laboratory, and the early computers at the University of Manchester may very well have been influenced by the work at Bletchley Park, even though any such influence could not be acknowledged.

Howard Aiken - Babbage's Dream Come True

It is worth allocating two or three lectures to a discussion of the electromechanical computing machines that were designed in the 1940's by Aiken and Stibitz and Zuse. Howard Aiken deserves credit for proposing (in 1937) the first large scale programmed computer that was actually built and put into productive use. Adequate credit should also be given to the IBM engineers who actually designed and built the IBM Automatic Sequence Controlled Calculator (ASCC), which is better known as the Harvard Mark I. It was just 100 years after Babbage proposed his Analytical Engine. It should be noted that Babbage's engines were designed to be mechanical, with the ultimate motive power provided by steam engines. The ASCC made use of the many advances in electromechanical technology in the punch card machines developed over a number of decades by inventors who worked for IBM. In England in 1946 Leslie J. Comrie wrote that the ASCC was "Babbage's dream come true". I once asked Aiken about this and he stated his opinion that Babbage was a dilettante. Aiken mentioned Babbage as a predecessor in his original proposal, and I think that he might have proposed a better machine if he had studied the Babbage design more carefully.

The Origins of Modern Computing

The climax of the first half of the course is our study of the origins of modern computing in the years between 1942 and 1946 at the Moore School of Electrical Engineering at the University of Pennsylvania, and the controversies surrounding the allocation of credit in that area. The first time I taught the course it came as a bit of a shock to me to realize that hardly any of the students knew anything about vacuum tubes, or had ever even seen one. It is interesting to note that the vacuum tube flip-flop was invented in 1919, but that its revolutionary effect on computing was not felt until the 1940's. The proposal to build the ENIAC that John Mauchly wrote in 1942 represents to me the start of the Computer Revolution. The motivation for the ENIAC was the calculation of trajectories, but it was designed to be a general purpose programmed computer. When it was finished it was orders of magnitude faster than any other digital computer, and it could do calculations that it would not have been possible to consider before it was available. I like to mention the calculation of π to 2037 decimal places that was done just for the fun of it on ENIAC over a weekend. I point out that when I was an undergraduate in 1937-41 π was known to only 707 decimal places, a value that had been published in 1873 by the English mathematician William Shanks after years of hand calculation. Much later, in 1945, it was found that only his first 526 decimal places were correct.

At this point we are ready to study the origin of the idea of the stored program electronic computer in the original design of the EDVAC in 1944-1945. It is important that students understand enough about computer architecture that they can appreciate the fact that the EDVAC as described in von Neumann's famous 1945 document is a computer in the most modern sense of the word. One of the very first programs for the EDVAC that von Neumann wrote was a sorting routine, not a mathematical program, and Goldstine records von Neumann's comment that the EDVAC was a truly universal machine.

My own article, "The Origins of Modern Computing" (See list of references) serves as a supplement to the textbooks in this area. I think that the controversy surrounding the invention of the stored programmed computer adds interest to this topic. I try to get the students to see the inventors, Eckert, Mauchly, von Neumann, and others as people, not just as scientists and engineers. At different times and to different extents they realized the importance of what they were doing, and they sought for themselves the recognition, the fame, and also the financial rewards that could be expected. Some of the basic ideas of the stored program computer were being discussed by the Moore School group before von Neumann joined them as a consultant. Yet it is clear that von Neumann made major contributions to the design of the EDVAC. He had responsibility for the specification of the instruction code, and he contributed some of the important ideas about address modification during the running of a program. Goldstine states that "He (von Neumann) was, among all members of the group at the Moore School, *the* indispensable one." I tell the students that even though Goldstine was there, perhaps *because* he was there, his opinions in this area need not be accepted as gospel. I argue that there would have been an EDVAC, though perhaps a somewhat different one, even if von Neumann had never become involved with the work at the Moore School. I state unequivocally that I consider Eckert and Mauchly the inventors of the computer, but I try to present other points of view.

The ENIAC Patent Case

I have left out any mention of Atanasoff here until now, but in the course there is a discussion of the ABC computer in its chronological order, just ahead of the ENIAC, and at that time there is mention made of Mauchly's visit to Atanasoff at Ames, Iowa in June of 1941. I leave my discussion of the ENIAC patent case until after the EDVAC discussion. This is suitable, since it was dissension about patents that led to the breakup of the Moore School computer activity in 1946.

My Computing Reviews article mentioned above serves as the text for the discussion of the ENIAC patent case, especially since neither the Goldstine nor the Williams text has any material on it. I think that the discussion of the patent case livens up the course. My article was published along with replies by people who have written books and papers and reviews about its subject matter. The students thus become aware of the existence of a number of radically different views about the same subjects. In this case at least, history is more than just an accumulation of facts about things that happened in the past. I present my own point of view, that the judicial process inevitably "produces a distortion of history, since the facts are filtered through depositions and cross examination by lawyers whose only goal is to strengthen their own case and to weaken that of their opponents."

Later in the course I bring up other patent disputes, between Forrester at MIT and Rajchman of RCA concerning the magnetic core memory, and between Noyce at Intel and Kilby at Texas Instruments concerning the integrated circuit. This topic was accentuated during the fall 1990 semester by the surprising news announcement that a hitherto unknown inventor named Hyatt had just been awarded a patent on the microprocessor, a patent for which he had first applied in December of 1970.

The Beginnings of the Computer Industry

One of the major themes for the second half of the course is the origin and growth of the computer industry. There is a great deal of drama in the story of the efforts of Eckert and Mauchly to establish a viable computer company and to develop the Univac. Their dealings with the National Bureau of Standards, their problems with security clearance, their gross underestimate of the time and the cost to produce the Univac, and the resulting takeover by Remington Rand. This permitted them to finish and deliver the Univac, which was a great machine for its time, but Remington Rand did not have the resources or the drive to become the leader of a great new industry, and they abdicated the leadership role to IBM.

It is hard to discuss the growth of the computer industry in the 1950's to the 1970's without putting a great deal of emphasis on IBM. The history of IBM's entry into the industry, and of its phenomenal success have been very well documented in a number of books. I am particularly impressed by two excellent recent volumes, "IBM's Early Computers" and "IBM's 360 and Early 370 Systems" which resulted from a computer history project within IBM. Between them there are over 1500 pages devoted to the history of IBM's computer developments, including some major software projects, up to about 1975. I try to avoid too much emphasis on IBM, but that is difficult since there is no comparable documentation for other computer companies. There is an interesting history of DEC with the somewhat misleading title "Computer Engineering A DEC View of Hardware Systems Design."

Software

I spend about two weeks of the course on the history of computer software. That is not nearly enough time to do justice to the subject, which could itself easily be the subject for a full semester course. I hand out copies of my historical survey that forms the first part of the book "Programming Languages and Systems". That article only goes up to the mid 1960's, but I don't know of a more modern survey that would be appropriate. There is an excellent article, *The Early Development of Programming Languages* by Donald E. Knuth and Luis Trabb Pardo in HCTC that I recommend to the students, but that article would be appropriate text material for a more detailed advanced study of the history of programming languages.

The Three Computer Generations

I hand out copies of my 1969 Computing Surveys article to serve as supplementary text material for the study of the first two computer generations. There is a nice delineation between these generations, the first based on vacuum tubes and the second based on the transistor. Once we get into the discussion of the first generation it becomes clear that it makes sense to talk of early and late first generation computers, with the distinction based on the memory system technology that was used. The introduction of large fast reliable magnetic core memories in the mid 1950's had a revolutionary impact on the large scale computer field.

It is appropriate to leave the discussion of the third computer generation until there has been some discussion of software. It is difficult to give a good characterization of the third computer generation. Some say that the integrated circuit characterizes the third generation in the same way that the transistor characterized the second. That creates a problem, since the IBM 360 when it was introduced in the mid 1960's used hybrid circuits which don't quite qualify as integrated circuits, and certainly the 360 belongs to the third generation. Also the Control Data 6000 and 7000 series computers that dominated the large scale scientific computing field from the mid 1960's to the mid 1970's certainly qualify as third generation computers, and they used discrete components. I suggest that the large third generation computers differed from second generation systems because of their ability to support disk-based multiprogramming operating systems. This is also a good time to discuss the time-sharing systems that provided a bridge between the computer generations. The group that developed the Comprehensive Time Sharing System at MIT became fascinated with the ideas of virtual memory introduced on the Manchester Atlas, and tried to get IBM to include virtual memory on the new 360. When IBM refused they gave up on IBM and got General Electric to produce such a machine, the GE645, and they got Bell Labs to join with them on the Multics project. IBM then committed to a virtual memory 360 model 67 that kept GE from getting many more orders for their 645, but it was too late to win back the Multics group. Bell Labs eventually gave up on Multics but their UNIX system, which is used by all of our students, owes a great deal to the earlier Multics effort. A lecture on the history of Unix is of special interest to the class.

Term Projects

It is desirable that students in a course do some creative work on their own. I therefore require that each student write a fairly long report or paper (I suggest a goal of 20 pages) on some area of the history of computing. This is meant to be a term project, with topics selected by shortly after the midterm, and papers submitted during the last week of the semester. I provide a long list of suggested topics, and also encourage students to write on other topics of their own choosing, so long as they are relevant to the history of computing. I warn students that they should check for availability of references before committing themselves to a topic, but I have no objection if more than one student selects a given topic. They are expected to work independently.

I tell the class that the project is optional, but that failure to submit a paper will result in a grade of 0 that will count as 20% of their semester grade. Just about everyone submits some kind of paper. Many of the student papers show unmistakable signs of having been done with minimum effort at the latest possible time, and I sometimes get a bit discouraged with them and consider dropping the project requirement. However, there are enough good papers, and some really interesting papers, to make me feel that the experience is worth while, especially for the better students.

Topics selected by students have been quite varied. There have been a number of papers, some quite good, on the history of computer graphics and on the history of artificial intelligence. Expert systems, vision, robotics, and computer chess are among the areas of artificial intelligence used. Two students wrote papers on the history of Unix. Histories of computer companies have included Burroughs, Univac, Apple, and a long paper on DEC by a student whose father had been employed by DEC for many years, and who has himself since gone to work for DEC. He had some sources of information that were out of the ordinary, as did a student who had spent the previous summer working for IBM at Boca Raton and wrote an interesting paper on the history of IBM's entry into the personal computer market. Some students selected topics having to do with the history of computing in the ancient world. Others wrote on the history of microprocessors, and on personal workstations. There were two interesting papers on aspects of computer crime, one of which had the provocative title, "A Concise History of Malicious Programming."

Comments and Evaluations

I have given the course in the history of computing in the fall semester of three successive years. Each time I gave the course there were a number of students who took a very active interest in the subject and who learned a good deal about it. One of the best students suggested very seriously that a course of this type should be required of all CS students. He and some of the other students were surprised when I told them that very few CS departments offered such a course at all. One student wrote to me (by E-mail), "I enjoyed your class--I think more than any other I've taken in the CS department. The readings were quite fascinating, and, at times, even exciting." Other students talked to me about their visits to museums we had mentioned in class, the Computer Museum in Boston, the computer exhibit in the Smithsonian, and even in one case the Science Museum in London.

As is the case in many universities, Purdue provides a mechanism for the evaluation of courses and instructors by the students in those courses. The evaluation form contains an area in which the students may add their own comments. The forms do not contain the student's name or any other identification, and as further encouragement for them to express themselves freely the students are told that the instructor will not see the forms before grades for the semester are all in. The forms are tabulated by computer, and the tabulation along with those forms that contain student comments are returned to the instructor after the end of the semester. Participation in the survey is voluntary, and students who are absent from class on the day of the survey are classified as non participants. In two semesters, fall 1989 and fall 1990, there were a total of 55 students who participated out of a total enrollment of 68. Of these 55 students 32 rated the course good and 14 rated it excellent. 27 rated the instructor good and 23 rated the instructor excellent. I don't think that I am being immodest when I suggest that to a great extent the good instructor ratings in this course are based on the fact that I have great knowledge of the field, and that I was around and to some extent participated in important parts of the history.

I do not take these ratings too seriously, though I would worry if they were very low. I find some of the comments made by the students interesting. I think that a number of students whose comments would have been more negative prefer not to make any comments, or not to participate in the evaluation at all. Some students showed great enthusiasm for the course. The E-mail message quoted above is an example.

Several students wished I had devoted less time to people like Babbage and Turing, and more time on more recent developments. A second semester to cover more recent history appeared in two suggestions. A few students who made generally favorable comments also stated that the course was boring at times. I don't know how to avoid that. I try to spice my lectures with interesting anecdotes and bits of computer folklore, but the basic tenor of the course is a serious presentation of historical information that I consider very interesting in its own right. It is hard to remain interesting through two 75-minute lectures each week for a whole semester.

Some students found the course boring, missed many lectures, and got little or nothing out of it. I don't question the right of any student to be uninterested in the history of computing but I wish they wouldn't sign up for my course. Unfortunately there were a number of students in each class whose only reason for taking the course was to get credit for a computer science course that contained no mathematics or theory, and that did not require the difficult and time-consuming programming projects that are assigned in many computer science courses. Some of these students were surprised to find that I meant the course to be a senior level course that required study and extensive reading. I would like to be able to limit registration in the course to students who are seriously interested in the subject, but that does not seem to be practical.

Lecture Notes

For many of the topics that I cover in the course I supplement the texts with my own lecture notes. I post these as text files on a computer on which all students in the class have accounts. They can be accessed from public and private terminals and workstations connected to the university network, and from home computers through the public phone system. A number of transparencies are used in the lecture that corresponds to a text file. Copies of the transparencies are available to the students in the library, where they can make their own copies if they wish.

In many of the text files I try to include material that does not appear in our textbooks, and that I hope will add interest to the subject under discussion. Some examples of this are in Appendix A which contains excerpts from a few of the text files that were posted in the fall of 1990.

Conclusion

The History of Computing course that I have offered at Purdue has been an interesting and stimulating experience for me, and for some of the students who took the course. Several students told me so, either in the course evaluation comments mentioned above or in personal conversation. Even so I do not expect the course to survive in the curriculum of the Computer Science department after my retirement, which is imminent. No other member of our department has shown any interest in offering the course. It does not make sense for anyone to offer such a course who is not well versed in the subject. That is true for any subject, but by now most of our faculty in computer science have themselves taken the standard courses that are offered in such a department as part of their own undergraduate or graduate studies. Even if they are not expert in a particular subject, they can usually do a reasonable job of teaching it by basing their course on a course they themselves have taken, and by following a standard textbook. I don't know of any computer science faculty members here or at any other major university who have taken a course in the history of computing offered in a computer science department. In every university I know of in which such a course has been offered it has been a project of just one faculty member who has become fascinated with the subject. I do not foresee a rapid expansion in the number of universities that will offer courses in the history of computing. I hope that this paper and its appendices will give some help and encouragement to others in Computer Science who may wish to undertake to teach such a course.

Appendix A

This appendix contains excerpts from a few of the files that were posted in the fall of 1990 in connection with the course on the history of computing at Purdue University.

From lecture notes on Charles Babbage.

It has been known for some time that there had been an earlier, rudimentary proposal for a difference engine published in Germany in 1786. There is some interesting information about this earlier proposal in:

Lindgren, Michael: Glory and Failure *The Difference Engines of Johann Müller, Charles Babbage, and George and Edvard Scheutz.* MIT Press, 1989. 414p Craig G. McKay (translator)

Johann Helfrich Müller (1746-1830) was an engineer in an artillery corps in Hesse (now part of Germany) until 1769. After that he engaged in various engineering and building occupations first in Giessen and then in Darmstadt.

Müller invented a calculating machine in 1782. It was completed in 1784. The calculating machine that he built used Leibniz wheels, and apparently it worked well. A 50 page booklet about his machines, written by Müller and edited by Philip Engel Klipstein was published in Frankfurt and in Mainz in 1786. The book was of course written in German.

The final section of that book talks of further inventions, and of an arithmetical printing machine. In that section, according to Lindgren, "it is made clear that Müller . . . had invented a difference engine. . ." It is clear that he had the concept, but there is no suggestion that he had actually built, or had even produced a detailed design of such a machine.

There was a copy of the Müller/Klipstein book in Babbage's library, though no indication of when it was obtained. Among Babbage's letters Lindgren found a five page translation of parts of the Müller/Klipstein book that John Herschel had done for Babbage. It included the material about the idea of a difference engine contained in that book. The translation is undated, but Lindgren thinks it may have been done before Babbage started to design a difference engine, or perhaps very shortly after that work started, and that some of Babbage's ideas included in the design of his difference engine may have been derived from Müller.

From Lecture notes on Alan Turing

Born: June 23, 1912 Died: June 7,1954

Recent long biography: Alan Turing the enigma. by Andrew Hodges; Simon and Schuster, New York 1983. Title is a play on words. A reference to Turing's work on breaking German ciphers that used the Enigma machine in WW-II. Hodges was active in the gay liberation movement in England, and this book puts a great deal of emphasis on this aspect of Turing's life.

Note also Alan M. Turing. by Sara Turing; Heffers, Cambridge 1959. A rather touching but probably incorrect biography by his mother. No mention of sexual preference. Suggests that he poisoned himself by accident, and doubts that it was suicide.

. . . During the next year (1935-36) he worked on his paper on computable numbers. The main result was a proof that there could be no process that could satisfy the requirements of Hilbert's third problem. He submitted his paper to Newman in April, 1936. In May, 1936 Newman received a copy of Church's new paper that included a solution of Hilbert's third problem using the Lambda Calculus. Again it was proved that no such process could exist. Church undoubtedly had priority as the first one to solve the problem, but Newman suggested to the London Mathematical Society that they should publish Turing's paper because of the fact that their methods were different and the problem was so important. Church, who reviewed the paper, agreed but suggested that Turing add a section to prove that his definition of *computable* was equivalent to Church's *effectively calculable*.
. . . . It is interesting to note that Emil Post published a formalism that seems almost identical to that of Turing at almost the same time. Post was aware of Church's work at the time he published his own, but it is almost certain that he was unaware of Turing.

From lecture notes on ONR and NBS

Note: In August 1988 the National Bureau of Standards (NBS) became the National Institute of Standards and Technology. The Office of Naval Research (ONR) was established by Congress in 1946. . . . Mina Rees states that ONR "played a vital role in the postwar support of basic research in the sciences at the nations universities . . . before the establishment of the National Science Foundation in 1950 and its expanded funding a few years later."

Mina Rees *The Computing Program of ONR, 1946-1953.* AHC Vol. 4 No. 2 April, 1982. Reprinted in CACM Vol. 30 No. 10 October, 1987. Mina Rees, Ph. D. Chicago, 1931. Organized the Mathematics Program of ONR in 1946. Director, Mathematical Sciences Div. 1949-1952.

Early in 1946 the Chief of Naval Research suggested that ONR and NBS jointly establish a national interagency computing center to be run by NBS. In September, 1946 ONR transferred $425,000 to NBS to assist in establishing a federal computing center, and to obtain a computer which is referred to as the ONR machine,. . . .

The Census Bureau had been approached by Eckert and Mauchly, and in the Spring of 1946 the Bureau asked NBS to purchase an electronic digital computer for census needs. For a 15% fee NBS would be responsible for selection of the supplier and for technical supervision of its construction. In October, 1946 the Census Bureau transferred $300,000 to NBS.
. . . . Applied Mathematics Executive Council (AMEC), an interagency council set up by NBS in January 1948. At its meeting on March 22, 1948 it considered two proposals. Eckert and Mauchly offered to build a machine for the Census for $150,000. Raytheon's price was $650,000 for a 4000-word-memory machine, and $425,000-595,000 with smaller memory. The AMEC decided to order three systems from Eckert-Mauchly. One was for the Census, one for the Air Materiel Command, and one for the NAML (the ONR machine).

Before any orders were executed it became known that the Eckert Mauchly company had been denied security clearance in connection with a bid to supply some electronic equipment to the

Navy.
See *Bit by Bit : An Illustrated History of Computers.* By Stan Augarten. Ticknor & Fields, New York 1984. An appendix contains a brief discussion of and some excerpts from the FBI dossier of John Mauchly. It points out that Mauchly was cleared for access to Restricted information in December 1952, and that he was finally granted Secret clearance in 1958. Augarten states that Mauchly had been "an innocent victim of anti-communist hysteria."

Defense agencies were being very careful. The AMC and ONR would prefer to do business elsewhere. This is probably the reason that Raytheon finally got the order for the ONR machine. Mina Rees refers to "irrelevant security considerations that plagued us at that time." . . . . On June 25, 1948 a single Univac was ordered from EMCC for the Census Bureau for $169,000. EMCC estimated development cost at $400,000. In October 1948 the Air Force Comptroller and the Army Map Service ordered Univacs in a supplemental agreement to the Census order.

Raytheon had no clearance problems, and was in much better financial condition than EMCC. Also, the Raytheon design had a number of attractive features. Mina Rees states that "diversification both of design and source of supply was highly desirable." Raytheon had an order to build two "Hurricane" computers for the Special Devices Center of the Navy. They offered to build one for NBS after the first two were completed. Contract for cost plus fixed fee, but not to exceed $350,000 was signed in January 1949. This was to be the ONR machine. The contract was eventually cancelled after Raytheon asked NBS to extend the delivery time and double the price. A single Hurricane, renamed Raydac, was installed at the Naval Air Missile Test Center, Pt. Mugu, Calif. in July 1952.

Harry Huskey returned from England to join the Electronics Division staff of NBS in January 1948. He suggested that NBS build an interim computer since no computers had yet been ordered, and delivery of the commercial computers might be far off. George Dantzig , who represented the Air Comptroller's Office on AMEC, had developed the Simplex method for Linear Programming by the end of 1947 and needed significant computing capability to use it on real problems. At the May 18, 1948 meeting of AMEC decision was made to build the interim computer. Samuel Alexander was to be in charge. This became SEAC, the Standards Eastern Automatic Computer. . . . . SEAC was dedicated June 20, 1950 and was the first stored program computer in production use in the U. S.

From lecture notes on the IBM 701 and 704

In February, 1951 Hurd and Birkenstock were authorized to obtain orders for the Defense Calculator at $8000 per month, with first delivery scheduled for December 1952.

The Defense Calculator which was later called the 701 was to be "like IAS but with good input-output." . . . . Memory size was to be 2048 words. "One reason for this rather large size, by present standards was the policy of replacing control circuits by subprograms . . ." Quote is from Buchholz: *The System Design of the IBM 701 Computer* in Proc. of the IRE, Vol 41, No. 10 October, 1953.
. . . In mid May 1951 they decided to stop soliciting orders for the system. By the time pending orders were processed they already had letters of intent for 27 systems. In June they

reported to management that the $8000 monthly rental estimate was too low. It would have to be $12,000-$15,000. In April, 1952 customers were told that systems costs would be from $11,900 to $17,600 per month. By August 1952 there were firm orders for 13 systems, plus a 14th to replace the SSEC in New York. A decision was then made to build a total of 18 systems.

... Near the end of 1953 Gene M. Amdahl was made head of a 701 enhancement project which soon developed into a project for the design of a follow-on computer. His group designed floating point arithmetic and index registers and many additional instructions. The IBM 704 was announced with CRT memory on May 7, 1954, but by October that was changed to magnetic core memory. Deliveries began late in 1955, and a 32k core memory was introduced in Spring of 1957. In all 123 704 systems were produced. The 704 gave IBM almost complete dominance of the Scientific Computing field.

Appendix B

Syllabus

CS 490R: History of Computers

Early History
- Al-Khowarizmi, the Abacus, Fibonacci
- John Napier, Napier's Bones, Logarithms, Henry Briggs
- Invention of the Slide Rule, Gunter and Oughtred

Early Mechanical Calculators
- Schickard, Pascal
- Leibniz, Charles Thomas of Colmar, others

Charles Babbage, Ada Augusta-Countess of Lovelace
- Difference Engine, Analytical Engine

Hollerith and Punch Card Machines, Powers, The early history of IBM

Alan Turing and his famous paper on Computable Numbers

Electromechanical Computers
- Howard Aiken, IBM ASCC and SSEC, Harvard Mark I and II
- George Stibitz and the Bell Relay Computers
- Vannevar Bush and the Differential Analyzer
- Konrad Zuse in Germany

The First Electronic Computers
- Cipher Machines - Colossus
- John Vincent Atanasoff and the Atanasoff-Berry Computer
- J.Presper Eckert and John Mauchly and the ENIAC
- The EDVAC and John von Neumann

Early First Generation Computers
- Computers in England
  - Turing's ACE, Pilot ACE, at NPL
  - The EDSAC - Maurice Wilkes, microprogramming
  - Kilburn and Williams and the Manchester Machines
- The Institute for Advanced Study Computer
  - IAS clones: Ordvac, Illiac, Weizac, et al.
- Whirlwind and SAGE
- ONR and NBS, SEAC and SWAC

The Beginnings of the Computer Industry
- Eckert-Mauchly Computer Corp.: Binac and Univac
- IBM Electronic Calculators and the CPC
- IBM 701 and 702, NORC
- RCA Bizmac, ERA(Univac) 1103

Magnetic Drum Computers
- Early Magnetic Drum Computers
- IBM 650, Burroughs(ElectroData) 205
- Univac Solid State Systems

Magnetic Core Memory Machines

        - IBM 704, 709, 705
        - Univac II, III, 1103a 1105
        - Honeywell Datamatic 1000, Burroughs 220
- Software Systems
    - The Beginnings of Software
    - Univac Systems, Grace Hopper and John Mauchly
        - Short Code, A2, A3(Math-Matic), B0(Flow-Matic)
        - GP(Fleximatic)
    - Interpretive Systems
        - Speedcode on the IBM 701, Bell System on the IBM 650
    - Assemblers
        - SOAP on the IBM 650
        - SAP, SAP II-VII on the IBM 704
    - Algebraic Compilers
        - Laning and Zierler at MIT
        - Fortran, John Backus et al.
        - Purdue Compiler, IT, Fortransit, Alan Perlis
        - Algol 58 dialects, Balgol, Neliac, Jovial, MAD, etc.
        - Algol 60, Backus Normal form, Peter Naur
        - Pascal, Algol 68
    - Business Oriented Languages
        - Predecessors to COBOL
            - Flow-Matic, AIMACO, Commercial Translator
            - FACT
        - COBOL
    - Early Operating Systems
- Second Generation Computers
    - Invention of the Transistor. Point-contact and Junction Transistors
    - First Transistorized Computers
        - IBM 608, RCA 501, NCR 304, IBM 7070
    - High Speed Scientific Computers
        - Surface Barrier Transistors and Drift Transistors
        - TX-0 and TX-2 at Lincoln Laboratories
        - Philco Transac S1000 (SOLO), S2000, Honeywell 800
        - Supercomputers, Stretch and Larc
        - IBM 7000 Series
    - Minicomputers, DEC and others
    - Time Sharing Systems
        - CTSS at MIT
        - GE 200 and Basic at Dartmouth, SDS 940 at Berkeley
- The Third Generation
    - Silicon Planar Transistors, Integrated Circuits, Hybrid Circuits
    - Atlas, Paging and Virtual Memory
    - Burroughs 5000, 5500, 6500, etc.
    - Control Data 6000, 7000 systems,
    - IBM 360, 370, PL/1
    - GE 635, 645, Multics, Unix

Supercomputers

- Illiac IV
- Star 100, Cyber 203, Cyber 205, ETA 10
- Cray I, Cray X-MP, Y-MP, CRAY II, CRAY III

Microcomputers

- Intel 4004, 8008, 8080, Motorola 6800, MOS 6502, Zilog Z-80
- Microcomputer kits, Altair, IMSAI and others
- Commodore PET, Tandy TRS-80, Apple I and II, Macintosh
- IBM PC and compatibles
- Workstations, Xerox PARC, Sun and others

Appendix C

References for The History of Computing

Annals of the History of Computing (AHC). A quarterly journal published since July, 1979.

The Origins of Digital Computers (Origins). 580p. Edited by Brian Randell. Springer-Verlag 1982. A collection of original documents written by major contributors to the computer field up to and including the development of the early digital computers.

A History of Computing Technology. By Michael R. Williams. 432p. Prentice-Hall 1985. A textbook based on a course given by the author in the Computer Science department at the University of Calgary.

A History of Computing in the Twentieth Century (HCTC). 659p. Edited by N. Metropolis, J. Howlett, and G.-C. Rota. Academic Press 1980. The proceedings of the International Research Conference on the History of Computing held at the Los Alamos Scientific Laboratory 10-15 June 1976.

The Computer from Pascal to Von Neumann. By Herman H. Goldstine. Princeton University Press 1972. 378p. A history written by one of the participants in the development of the most important of the early electronic digital computers.

The Origins of Modern Computing. by Saul Rosen. with responses by Herman H. Goldstine, Alice and Arthur Burks, John V. Atanasoff II, Nancy Stern, Bernard A. Galler, Clark R Mollenhoff, Alan R Mackenzie, and Eric A Weiss. Computing Reviews, Vol. 31 No. 9 (Sept. 1990) p449-481.

Bit by Bit : An Illustrated History of Computers. By Stan Augarten. Ticknor & Fields, New York 1984. 324p. An interesting history by an author whose background is in history and in journalism.

Portraits in Silicon. By Robert Slater. MIT Press 1987. 374 pages. Short biographical sketches of "the people behind the computers." Includes information from personal interviews where possible. Interesting. The author is a professional writer not a historian or a computer professional.

Morrison P. and Morrison E. Ed.(1961) Charles Babbage and his Calculating Engines, Dover Publications, New York 1961. 400p. Look under Babbage in the library. Anthony Hyman: *Charles Babbage Pioneer of the Computer.* Princeton University Press, Princeton N.J. 1982. 287p. The first is an anthology of writings by and about Babbage. The second is a good modern biography of Babbage.

Alan Turing the enigma. by Andrew Hodges. Simon and Schuster, New York 1983, 587p. An interesting biography. Title is a play on words. A reference to Turing's work on breaking German ciphers that used the Enigma machine in WW-II.

Memoirs of a Computer Pioneer. By Maurice V. Wilkes. MIT Press, 1985. 240p. Wilkes has been active in the computer field, mostly at Cambridge U. in England, since 1946. A very interesting autobiography.

Austrian, Geoffrey D. *Herman Hollerith Forgotten Giant of Information Processing.* Columbia University Press New York 1982. 418p. An interesting biography of Hollerith, the inventor of punch card machines.

IBM's Early Computers. By Charles J. Bashe, Lyle R. Johnson , John H. Palmer, and Emerson W. Pugh . 716p. The MIT Press 1986. IBM's 360 and Early 370 Systems. By Emerson W. Pugh, Lyle R. Johnson, and John H. Palmer. 819p. The MIT Press 1991. Two excellent long books by people who were themselves involved in many developments at IBM and who had access to company records and documents.

From ENIAC to UNIVAC: An appraisal of the Eckert-Mauchly Computers. By Nancy Stern. 286p. Digital Press 1981. A Ph. D. dissertation in the History of Technology. Note that an appendix of this book provides the first publication in full of Von Neumann's "First Draft of a Report on the EDVAC"

Reckoners : The Prehistory of the Digital Computer, from Relays to the Stored Program Concept, 1935-1945. by Paul E. Ceruzzi. Greenwood press, Westport, Connecticut 1983. 181p. A historian discusses the work of Zuse in Germany, The ASCC at Harvard, the relay computers at Bell Labs, and the Eniac, all in some detail.

Alice R. Burks and Arthur W. Burks. *The First Electronic Computer- The Atanasoff Story,* The University of Michigan Press 1988. 387p. Clark R. Mollenhoff. *ATANASOFF Forgotten Father of the Computer.* Iowa State University Press. 1988. 275p. These two books, one by a participant and the other by a journalist, are part of an attempt to revise the early history of electronic computing, and to give Atanasoff credit for the inventions made by Eckert and Mauchly.

History of Programming Languages. Edited by Richard L. Wexelblat. Academic Press 1981. 758p. Proceedings of the ACM SIGPLAN History of Programming Languages Conference. 1-3 June 1978, Los Angeles, California. Reports by the people who were involved in the design and development of most of the important first generation of programming languages.

Programming Languages: History and Fundamentals. by Jean E. Sammet. Prentice-Hall 1969 785p. An excellent survey of the major programming language developments up to the mid 1960s.

Programming Systems and Languages. Edited by Saul Rosen. McGraw-Hill 1967. 734p. A collection of articles, some of which have historical interest. The book starts with a historical survey by the editor. *Electronic Computers: A Historical Survey.* by Saul Rosen. Computing Surveys, Vol. 1, No. 1. March 1969. p7-36. The only Journal article included in this list of books.

Computer Structures: Readings and Examples. Edited by C. Gordon Bell and Allan Newell.

McGraw-Hill 1971. 668p. A very large book. It contains many papers of historical importance. A revised edition was published in 1982 with C. Siewiorek listed as a third editor.

Project Whirlwind The History of a Pioneer Computer by Kent C. Redmond and Thomas M. Smith. Digital Press 1980. 280p.
A history of one of the most important of the early computers.

Papers of John von Neumann on Computing and Computer Theory. Edited by William Aspray and Arthur Burks. MIT Press 1987. 624p. Contains a number of the most important papers in the early history of computing.

From Dits to Bits: A personal history of the electronic computer. By Herman Lukoff. Robotics Press 1979. 219p. An informal autobiography of an engineer who worked on the ENIAC, then joined Eckert and Mauchly when they set up their first company in 1946 and played an important part as an engineer for Univac I and for other Univac computers.

Computer Engineering - A DEC View of Hardware Systems Design. by C. Gordon Bell, J. Craig Mudge, and John E. McNamara. Digital Press 1978. 585 p. The title is misleading. This is an excellent engineering oriented history of DEC computers. Most of the book consists of a collection of previously published articles and technical reports.

A few Good Men from Univac. David E. Lundstrom. The MIT Press 1987. 227p. A history of Univac in St. Paul and of Control Data Corporation from the point of view of an engineer who worked for one and then the other.

Paul Freiberger and Michael Swaine **FIRE IN THE VALLEY** *The Making of the Personal Computer*. Osborne/McGraw-Hill 1984. 288p. An interesting and dramatic story of the origins and early development of the personal computer.