Memories of the Bureau of Standards' SEAC

RALPH J. SLUTZ

During the late 1940s, Johnny von Neumann gave several lectures describing the computers that were being developed then. I've always remembered a comparison he gave to make the power of the computers more understandable. He compared 15 minutes of running a particular computer to hiring a group of some 20 people using desk calculators, giving them their instructions, and locking the door on them for about a year—then finding out what went wrong because you had overlooked something in your instructions.

Lest we forget what computation of that sort was like, I shall give a brief excerpt from the memoirs of an early member of the WPA Mathematical Tables Project in New York City. This project was established in 1938 with a very small number of mathematicians and a fairly large number of people from the relief rolls of the Depression. The people who came from the relief rolls were a varied sort, and more than rusty in their arithmetic. Therefore, they were divided into four different groups. One group was to do addition, a second group to do subtraction, a third to do multiplication, and the fourth group to do division and check the results. The people were installed in an abandoned stable in New York City, and the four groups were seated facing the four walls of the room. Each group faced a wall on which there was a large poster giving them their most important instructions, and to keep negative numbers straight each person was given both black and red pencils to work with. The poster giving instructions for the addition group said:

ISBN 0-12-491650-3

Black plus black makes black.
Red plus red makes red.
Black plus red or red plus black,
  hand the sheets to group 2.

Ten years later, this humble beginning had grown into the National Applied Mathematics Laboratories of the Bureau of Standards under Dr. John Curtiss. Also in the Bureau of Standards was the Electronic Computers Laboratory, under Sam Alexander, which was concerned with the hardware side. At this point I should like particularly to remember Sam Alexander, who was the moving spirit in the Bureau's hardware work on computers, and who should be writing this paper if he were still alive. He always treated me more like a son than an employee, and we had a warm and productive relationship. Sam's group in 1948 had developed several computer components, particularly involving pentode vacuum tubes and pulse transformers. The group had also developed some of the input-output equipment for the Institute for Advanced Study computer. Sam was also acting as technical representative and contracting agent for three to five major computers.

When I joined the group in 1948, computer development everywhere was being ruled by the "von Neumann constant."* Some people have mentioned it as being a part of von Neumann's machine, but as I remember it Johnny developed it as a universal constant—that constant number of months from now until everyone's machine is expected to be completed.

After the ENIAC, many computers had been started, but at that time none had reached the promised land. In fact, the Office of the Air Comptroller of the U.S. Air Force had a major contract for the development of a machine to be used for linear programming analyses of economic problems. This was largely sparked by George Dantzig, who eventually became so impatient with the von Neumann constant that he persuaded the Air Force to contract with the Bureau of Standards to develop a computer with a minimum of complications and quickly, to fill the gap. During the development phase, this was known as the National Bureau of Standards Interim Computer. When it became operational it was renamed the SEAC, for Standards Eastern Automatic Computer, but we who worked on it sometimes prefer the name Interim, because that shaped much of its design. Reliability was of course paramount in the design goals, but simplicity came next in order to reduce the development time. No provision was made for floating-point arithmetic, none for alphabetic input or output, none for magnetic tapes. Instead, the original goal called for input from Teletype paper tape, integer arithmetic throughout, and output to the same Teletype printer that had read the paper tape. At that time one of the most nearly operational memories was the mercury delay line design of the EDVAC project. So with the fine cooperation of the EDVAC people it was possible to purchase from the

---

* Known in Britain as the Hartree constant [*Editor's Note*].

EDVAC supplier a cabinet containing the mercury lines and the temperature control system. This of course determined the general shape of the machine. It would be of the general EDVAC structure with delay line memory and synchronous circuitry, but the circuitry in which the Bureau of Standards was experienced was very different from the circuitry being used by the EDVAC group. Thus the circuitry and the detailed logic were worked out entirely independently by the Bureau.

The logic design included integer multiplication, integer division, and logical multiplication. The hardware had two rather unusual features. One of these was that, except for the memory circuits, all of the pulse amplification was performed by a single type of pentode vacuum tube driving step-down pulse transformers. The pentodes gave a well-defined voltage output and relatively low inpedance because of their saturation characteristics. The signal then went to a 5:1 step-down pulse transformer, so the already low output impedance was lowered by an additional factor of 25. Thus the signal lines were of very low impedance and could drive as many as a dozen or more gates as well as long connecting lines without worrying about distributed capacitance.

The other outstanding feature was that all of the logic throughout the machine was done by germanium diodes. This, as I understand it, makes it the first computer to do all of its logic with solid-state devices. It did have vacuum tubes, but they were used solely for amplification while all gating, all clocking, all pulse reshaping were done by germanium diodes.

For short delays, wire-wound delay lines were used, so that a flip-flop was a 1-μsec circulating loop. This made the design and debugging a bit difficult, since there were no static states, but everything was pulsing at a microsecond rate. However, Julian Bigelow at the Institute for Advanced Study project had trained me carefully in the principles of tolerance analysis and its fundamental importance. At the Bureau of Standards we were able to apply these principles so successfully that we achieved a complete separation of electrical and logical design. Alan Leiner, who played the principal part in the final logical design, was able to work with a complete set of building blocks which had well-defined limits. He didn't need to worry at all about the electronics within the building blocks—as long as they were fitted together within timing tolerances and load tolerances, they worked.

This resulted in circuitry which had some 750 vacuum tubes and 10,000 germanium diodes. Of course, this was at a time when there was a tremendous worry about the reliability of vacuum tubes. The program planners had to have frequent reports of the total number of vacuum tubes in the design, but somehow they didn't worry about germanium diodes. It was easier to add 1000 germanium diodes to the design than to add 10 vacuum tubes. Using so many diodes was something of a gamble, for the diodes were a very new device at the time, and we used the old 1N34 whisker diodes. But the gamble paid off. After the machine was debugged, we set up a preventive maintenance schedule that removed every tube and diode from the machine and tested it individually at least once a month. (The diodes were all in plug-in clusters.) With this schedule the failures that were not caught during maintenance time were kept low—each month during computation time we had about one vacuum tube and about two germanium diodes go bad. Thus the balance of reliability was pretty good.

We actually had much more trouble from bad solder joints than we ever had from vacuum tubes, diodes, or delay lines. I can well remember that we established two standard debugging techniques. After about two hours a day of preventive maintenance, we would start a test program running. Then we applied the "stir with a wooden spoon" technique, which consisted of taking something like a wooden spoon and going around the computer, tapping everything you could see. If the test program stopped, you had found something. When that test was finally passed, we applied the Bureau of Standards' "standard jump." We were in a building with wooden floors that were not difficult to shake, so the standard jump consisted of jumping up in the air about 15 cm and coming down on the floor as hard as possible. If that test was passed, the machine was ready to tackle a computational program—and even more interesting bugs would show up.

Three particular bugs stand out in my mind. The first was in arithmetic. When the machine was nearly complete and we started running programs we found indeed that 1 + 1 = 2 and 2 + 2 = 4, but 4 − 2 gave different answers depending on what program was used! Some programs gave 2, while other programs gave 3. After considerable sweating we found that 4 − 2 = 3 whenever the *addresses* of the two numbers differed by 7(modulo 8). Excess carry bits from complementing the negative number were propagating through the three guard bits at the end of the word and carrying into the answer. We changed one wire and so redefined the arithmetic to be more acceptable by human standards.

The first significant problem was done in May 1950, concerning tracing skew rays in optics. This type of computation was used very extensively by the Bureau of Standards for many years. Steady work was also started on linear programming, but it was not many weeks before Nick Metropolis and Bob Richtmyer showed up with a problem from Los Alamos that they wanted to put on the machine. I didn't realize until much later how much of a habit it was for Nick Metropolis to show up with a problem as soon as somebody got a computer running. Anyway, they were perfectly willing to take any time that we would give them, so they got the time from midnight to breakfast. But although the machine was running many problems for the Bureau during the daytime, it just would not run Bob Richtmyer's problem at night. There was trouble after trouble after trouble, and we all took turns being the nighttime debugger. Finally at 4 a.m. one morning I discovered the cause. Since Bob and Nick were real mathematicians, they didn't do things like everyone else and write their code from the bottom of memory up—

they wrote it from the top down, which was possible since SEAC was a four-address machine at that time. This resulted in the instruction word having nearly all ones, instead of the more usual nearly all zeros. And a design error of 0.1 μsec in one of the delay lines of the instruction register caused an occasional bit to be lost when the register was nearly full. So at 4 a.m. I hung on (with loose wires) an additional delay line of 0.1 μsec and it stayed there for about five years until SEAC was moved to another location and so refurbished.

The third flaw was a real hardware flaw. We had a narrow walkway through the middle of the machine and a narrow door for access. In mid-1951 we were having an open house, and my wife came with my children. The machine was open, there were people everywhere, and everything was happening at once, when my wife came hurrying to me in the next room. Our two-year-old son was crawling along the interior walkway toward the power supplies, which were uninsulated since they were inside the cabinet. And my wife couldn't get through the narrow door to rescue him because she was pregnant with our next child.

Well, engineers never seem to be willing to let well enough alone. The original design was four-address, with the fourth address being the location of the next instruction so you could do optimum coding with respect to the timing of the mercury delay lines. However, this meant that the addresses split hexadecimal digits in the input, and so different addresses had different representations. It also meant that there were only four bits to use for instructions, and so there could be no more than 16 instructions. It also meant that the memory would be forever limited to 1024 words, but we engineers had grandiose ideas of putting on a lot more memory—perhaps even as many as 2000 words. So during the design we worked out a scheme of including about seven additional tubes and 10 toggle switches, which could change it to either a four-address or a three-address machine. For some years it would run half of each day in the four-address mode on computational problems, and then the switches were flipped and it ran the other half-day in the three-address mode for testing out additional memories.

With these provisions for future growth, additional instructions were included permitting the use of two instruction counters and permitting the addresses to be relative to either of the instruction counters, thus creating a convenient subroutine jump and return. Other additions were magnetic wire and tape input-output devices. The first was a magnetic wire cartridge that was part of a commercial dictating machine. It took about a second for the wire to come up to speed, so it was only practical for handling "large" records of a hundred or more words at a time, but by leaving gaps of several seconds between such records it was possible to use one cartridge for many records. We then went on to add magnetic tape drives that avoided the problem of servoing the magnetic tape reels by getting rid of the reels. The tape was stored in a kind of two-dimensional wastebasket consisting of a pair of parallel glass plates. The tape was just dumped between the glass plates and it folded loosely any way it wished at the bottom. Three such plates made two wastebaskets for the two ends of the tape, and it worked very nicely for up to about 200 ft of tape, although we did have to use either metallized tape backing or radioactive emitters to discharge the static electricity which built up. The only moving parts were two pinch rollers, one for forward and the other for backward motion.

Another major addition was a set of 45 Williams' tubes for additional and parallel memory of 512 words. Near the end of 1950, a cathode-ray tube (CRT) graphical output was added with a joy stick to simulate aircraft guidance and control. We simulated a radar display by a program in the computer, and used the joy stick to guide a second plane for interception.

During the first year we ran some 50 programs. I have mentioned three: linear programming, optical lenses, and Richtmyer's Los Alamos program. Others of significance that should be mentioned were tables for Loran navigation, optimum statistical sampling plans, wave functions of the helium atom, and design features of a proton synchrotron.

This being a historical volume, I should perhaps say a little something about the historical position of the machine. For myself, I like to think of it as the first machine in this country to break the von Neumann constant. It was demonstrated in April 1950, and was in full mathematical use by May 1950. Thus I suppose you could say it was the first of the von Neumann type, or stored-program type, of computers in this country. But we recognize full well that in England the EDSAC, the Manchester machine, and perhaps the Pilot ACE can lay prior claim on an international basis.

I had thought that SEAC was the first with a remote terminal. In the middle of 1950 we wanted to demonstrate it to the people of the Bureau of Standards, and so put a Teletype a quarter-mile away in a lecture room much larger than the limited space around the computer. A few months later we put a terminal in a Washington hotel several miles away for a general conference. But I have heard that George Stibitz put a remote terminal on the relay computer long before. I do think it was the first to do all of its logic with solid state devices, and perhaps it was the first with cathode-ray tube graphical output.

However, much more important than any enumeration of its firsts is the question of whether it fulfilled its design goals, and I think it did. It was designed as the NBS Interim Computer. It grew up to get the less transient name of SEAC, and for a few years it satisfied its design requirement by providing a very significant part of the total United States's computing capacity. In addition to its original design goal, however, we were able to put enough growth potential in it to permit expansion in its instruction set, its memory (to 1500 words), and its input-output devices. With these expansions it went on to be a workhorse computer for the Bureau of Standards for a total of 14 years.

Finally in 1964, 14 years after it came to life, the once shiny new machine could still be polished up to be shiny, could compute far better then when it was new, but was hopelessly outclassed by its successors. But I think the attitude of the many of us who worked on it can well be given by a short poem that Ida Rhodes wrote for *Datamation* magazine at the time that it was retired. She said:

> Say it's weary, say it's slow,
> Say that luxury ignored it;
> Say it's growing old, but know
> We all adored it.

ENVIRONMENTAL RESEARCH LABORATORIES
NATIONAL OCEANOGRAPHIC AND ATMOSPHERIC ADMINISTRATION
BOULDER, COLORADO