ALSO SPRACH VON NEUMANN

by Eric Blodax
with illustrations
by
Stew Burgess

PROLOGUE: When the editor of DATAMATION suggested that I write a nostalgic series on the early days of digital computing, I thought it was a dumb idea. In fact, I was insulted. After all, the editor referred to me as a "genuine old-timer." I know I'm genuine, but I'm not that old.

However, after honest contemplation, I must admit that I have been around "computing," whatever that may be, for a long time. I've been in the machine room, and the keypunch room, and the programming room, and the logic design room. I've been in the circuit board room and the solder dip room, and in just about every other room with occasionally too little time to make the bathroom.

I did not work on the Harvard Mark I, as some of my friends did. But I did work with some of the squirreliest contraptions ever called computers. It may be, in fact, that a cross section recounting of how it used to be may help put some of today's wet-eared, callow kids in their proper place. They may even realize how soft they have it with their preprinted coding forms and their big paychecks.

This is *apologia in toto,* if any be required or excusable, for what follows.

It all began by accident. There is no sensible reason that I should have been swept up into the computing business. I might have had a perfectly respectable job in used car sales or in burglary, but it was not to be. I was actually enjoying a perfectly fine job as a junior engineer, grade B, in the Airship Foundry. I was in full charge of a small group of sweaty ladies who did engineering calculations by poking at a room full of rickety Marchants[1].

Then Old Robert came along. I call him old because he must have been nearly 30 at the time. Besides, I change his name as I frequently do in this opus in order to protect the guilty.

Old Robert was a new, hot-rock hire out of the Competing Aircraft Corporation of America. Competing Aircraft exercised its advertised leadership by being way out front in the application of computing to new design glitches. Robert had become disgruntled when some ten-dollar-a-day administrator misplaced his parking pass, and he had to walk into the plant from Glendale. Robert was thus gleeful when he was hired into our Way-Out-Research Staff by Dr. Attenuator, our beloved and fearless old leader. Robert arrived at the Airship Foundry in due time to teach all us clods how it really was out in the modern world.

Robert promptly recruited me to become the first "programmer" in the place. This was an obvious wise move on

1 **Marchant: A now-extinct, sickly-green model of an electromechanical desk calculator, push-button actuated.**

Robert's part, since it promptly got him out of the programmer class and made him a supervisor.

The scientific equipment, most of which we "borrowed" from the accounting people, was overwhelming. We had a machine called the "IBM 603 Calculating Punch." It looked like a suitcase cabled to a coffee grinder, or, possibly, a milking machine. You could see giant tubes glowing through the slits in the suitcase.

The opcode set of the 603 was a trifle limited. It could add and subtract and multiply, but it couldn't divide. We had to go back to the Marchant to calculate reciprocals, then keypunch these into cards so the 603 could read them. The necessity of doing this made engineering use of the 603 somewhat difficult.

Memory capacity of the 603 was also a bit small, since it had exactly zero memory locations. Programs were held in the wires of a tiny, enigmatic IBM plugboard.

Since mean time between failures of the 603 was very short, we were never able to complete a problem with it; thus it is difficult to assess, in retrospect, exactly what we might have been able to do with it.

Mostly we ran our tiny but important problems on the IBM 602A. To use this machine, one had to be a competent keypuncher, board wirer, and control-bar set-up man. However, the 602A was a massively engineered, highly reliable, electromechanical machine, and therefore it does not count among the computers. Everybody knows you cannot build a mechanical computer. It must be electronic.

The full dawn of the electronic age arrived in our little shop with delivery of our IBM 604. This machine represented a truly remarkable advance in the computing art. While it had no memory, it was magnificently programmable in three-address mode, provided one limited the program to no more than 60 instructions and plugged these into a plastic board in a metal frame. There was, however, a trivial I/O restriction: All input *and* all output had to be punched into a single 80-column card.

In spite of the minor limitations of the 604, we did a good deal of vital work with it. We waded into structures problems, solved the world of trajectories in at least two dimensions, and sailed easily through long equations in applied physics, like:

$$f = ma$$

Without question there were earth-shaking events in those days. Reaching into my bookshelf I take down my dusty copy of the proceedings of the Computation Seminar of December, 1949, held in the IBM Department of Education, Endicott, N. Y. One hundred and seven research engineers and scientists attended this seminar. It was staged by Dr. Cuthbert C. Hurd, director of the IBM Applied Science Department.

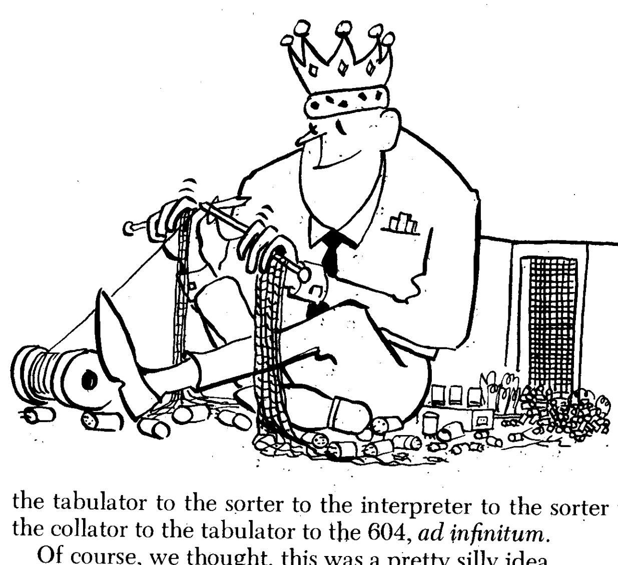
The keynote speaker was Dr. John von Neumann, who spoke on "The Future of High-Speed Computing." Dr. von Neumann addressed himself to a major concern of the day: the worry that computing machines would become so fast that they would rapidly make themselves obsolete, having solved all possible problems. Dr. von Neumann did not believe this fear to be based on fact. He admitted that, for problems then current, planning and coding required much more time than the actual solution of the problem. Yet, with clear vision, he foresaw the day when, since problem size was, in a very real sense, dictated by machine speed, the magnitude of problems would automatically increase so that problem solution time would become longer than planning and coding time.

At this same seminar, Dr. Richard W. Hamming of the Bell Telephone Laboratories spoke regarding some methods of solving hyperbolic and parabolic differential equations.

There were learned papers on sundry topics, including "Inversion of an Alternate Matrix," "The Construction of Tables," "An Algorithm for Fitting a Polynomial Through N Given Points," "A Punched Card Application of Monte Carlo Method," and "Remarks on the IBM Relay Calculator."

Doubtless all 107 attendees went home convinced of the secure future of their new-found profession. After all, the world had never before seen such an overwhelming attendance at a meeting set up for the discussion of digital computing.

Meanwhile, back at the Airship Foundry, word kept leaking in to us that out at Northrop a bright engineer named King Barley had spread the innards of his 604 all over the floor while he tried to substitute wires for the physical act of walking with cards in hand from the 604 to the tabulator to the sorter to the interpreter to the sorter to the collator to the tabulator to the 604, *ad infinitum*.



Of course, we thought, this was a pretty silly idea.

Shortly thereafter, IBM announced the CPC, the aboriginal three-letter mnemonic, standing for Card Programmed Calculator. We ordered several.

This impressive collection of cabled-together boxes included a 604, a tabulator, and a bunch of electromechanical storage devices which looked suspiciously like hastily reworked "accumulators" from inside a tabulator. These memory locations came twelve in a box, and the customer could select up to three boxes, giving him the unheard-of capacity of 36 memory locations, each consisting of 10 decimal digits, plus sign. To access these memory locations one had, however, to endure a computing delay of one or more cycles (tabulator turns) of time.

A great breakthrough had been accomplished. No longer was the sole input and output capability found in the delayed passage of one card through the reader-punch of the 604. Now we had a *real* I/O machine: the 402 tabulator. At least the number of cards was unlimited. The 402 would read them all day at some high percentage of its rated speed. With each reading the magnificent type bars rose and fell, some alphanumeric and some numeric. They looked like the tide ebbing and advancing with the phases of the moon, and they were damn near as fast.

The CPC shortly became the only machine of importance to people in a hurry.

By people in a hurry, we mean those up tight to see Dr. von Neumann's predictions fulfilled, not those anxious to make a buck on their fulfillment.

totus mundis histrioniam agit

While our daily work of calculating engineering structures, strictures, and sutures went on at the Airship Foundry, massive rumblings began to be heard in our new world of computerdom. The U.S. federal government, in a vast, precedent-setting move, acting through its duly constituted, knowledgeable, and contemplative authorities, had caused to be negotiated a contract with the Eckert-Mauchly Corp. of Philadelphia—a contract to supply a really big computer to the Bureau of the Census. Not since the decreee of Caesar Augustus that the world should be numbered was there government action triggering off such a monumental series of events.

The government had financed computing machines before, and has done so many times after, but there was something about the genesis of Univac I that marked the dawn of a new age. No doubt the contract for the first Univac, like many contracts negotiated by well-meaning government committees, was optimistic in delivery and performance schedules, and deficient in price, for federal government procurement is marked by errors that a housewife knows how to avoid. Yet this was a beginning.

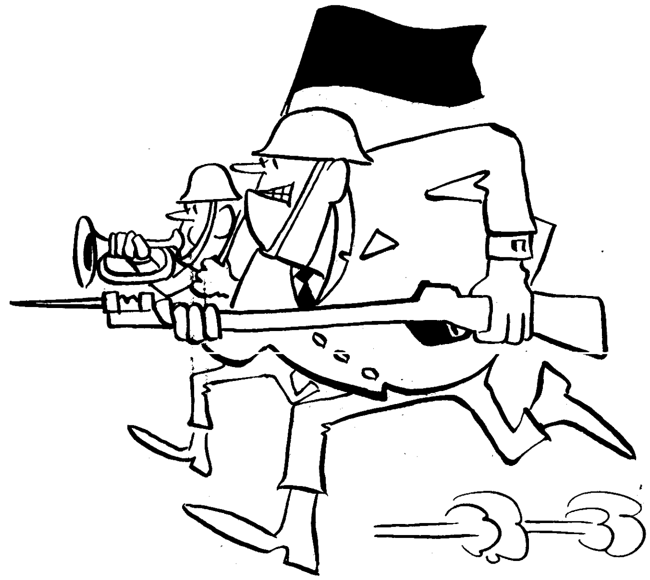
Ere long the small Eckert-Mauchly Corp. had become the Univac Division of the (then) Remington Rand Corp. Progress went on apace, and Remington Rand loomed as the only American company with a big, high-performance electronic computer on the boards and with production scheduled. High-quota adding machine, shaver, and tab equipment salesmen boned up to sell Univacs to the world.

IBM, which had sold Hollerith (square-hole) card systems to almost the level of a national standard, was nearly caught napping, but not quite. It was the time of the Korean War and reimposed priorities, and the Defense Calculator project (later the IBM 701) got into high gear in a hurry.

Well-trained teams of IBM Applied Science men and marketeers (all IBM representatives are, by definition, well-trained) covered the country, making their pitch particularly to the engineering/scientific types. Though they ran behind in timing, there was no deficiency in their push or pitch.

Whatever the reasons, lease vs. sale or you-name-it, the Univac troops were repeatedly able to snatch defeat from the jaws of victory. They made some impressive sales, particularly to such customers as insurance companies, but somehow still managed to end up in second place when the dust settled. At the Airship Foundry we ordered a Defense Calculator, and proceeded to prepare a building just to house it.

The contrasts between the two computers are interesting, and even ludicrous from today's point of view. The Univac was a decimal, serial machine with dual, voting circuitry. The 701 was binary and parallel. I/O was copious and varied on the Univac and limited to cards on the 701 (since no one of that day had any way to prepare a 701 tape off-



line). The Univac was pitched as a "business-oriented" machine, while the 701 was alleged to be "engineering/scientific oriented."

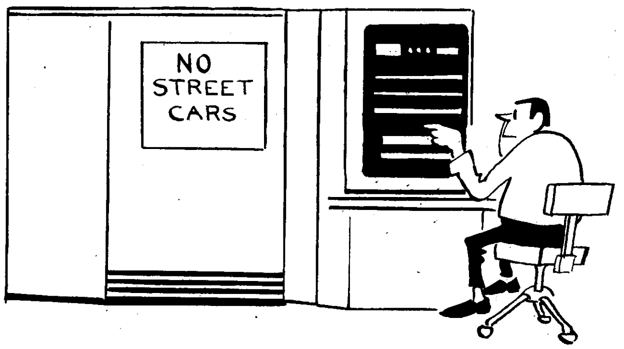
The main memory systems of the two computers presented one of the most fascinating of the contrasts. Univac memory consisted of a set of cylindrical tanks studded with mountings for piezo-electric crystals. The write crystal put a never-ending string of sound pulses into the mercury which filled the tank. The read crystal opposite read this string and sent it to the amplifier to be reshaped and rewritten by the write crystal. Thus memory was a vast assemblage of acoustic delay lines. Univac I memory was therefore susceptible to interference from such effects as loud door slams. IBM 701 memory, on the other hand, was electrostatic. It wrote bit patterns on Williams tubes (3-inch cathode ray tubes) and thus had continually to regenerate as the electrostatic charges faded. This memory was susceptible to interference from opening circuit breakers and passing streetcars. Both memories in their original installations exhibited a mean time between failures so short as to be barely measurable.

Actually the Univac I suffered from being ahead of its time. Its I/O devices were elegant, magnificent in concept, expensive, and unreliable. They offered the user capabilities that were unheard of then but commonplace today.

When our 701 (Defense Calculator) was delivered to the Airship Foundry we were ready for it. We had read the installation instructions carefully, and had become expert in the care and feeding of a big computer. Working hand-in-foot with our brilliant plant engineering crew, we had procured a giant motor generator set to power the computer. This machine was approximately the size of the left bank powerhouse at Grand Coulee Dam. The plant engineers had tenderly hung a great ice machine in the building overhead, and had designed a perforated ceiling that would let the cold air extrude gently down over the entire 701.

The computing room ceiling thus looked like the world's largest spaghetti die, but it did spill frosty, filtered air at a great rate.

It took a week or so to get the 701 set in place on its brand new false floor, and to string the subterranean cables that hooked all its brains together. This task was completed on a Friday and, after a long coordinating session with the IBM field engineers, we determined that it would be a good idea to turn on the great ice machine to cool the computer



over the week end so it would be ready to turn on Monday morning in a blaze of glory.

As fortune would have it, the week end proved to be a long and exceptionally rainy one. When we came into the new computer facility on Monday morning the rain was still pouring down.

The great ice machine was still humming smoothly, and the Coulee Dam generator looked ready to spring into action. But a miracle had been wrought over the weekend. The 701 room had turned into a Christmas scene. Snow was gently falling from the spaghetti extruders in the ceiling, and it blanketed the computer, the false floor, and the entire 701 to a depth of, say, half an inch.

It took the IBM engineers about a week to dismantle the 701, dry it out thoroughly, and put it back together. This interval gave the plant engineering crew time to redesign the control system of the ice machine.

And so the great day came when the Coulee Dam generator was turned on to fill the world with its 60-cycle hum. The Williams tubes ignited with a glowing pattern of confused blue dots, and the card reader went pockety-gleep. The 701 was on the air.

We spent several weeks learning how to keypunch in binary, and trying to find out why a mag tape written on one servo would not read on any other, but gradually we learned to live with our new, magnificent computer. It was time to invent software systems.

For nearly a year we had a study team assigned to play with and design a "programming system" for the 701. The machine was delivered in a most virgin state, understanding only its own opcode set and nothing else. There was a "machine manual," since systems manuals had not been thought of yet. This was written by Sid Lida of IBM, and it was a beautiful job, possibly the finest such manual ever written. The writing of manuals by committee had not yet been invented.

In spite of our year of software planning, getting the 701 on the air proved to be equivalent to enticing a sick camel to stand up under a two-ton load. We worked under the "clean machine" concept in which no software was ever resident, but the entire facilities of the computer were available to the next clod trying to add two and two.

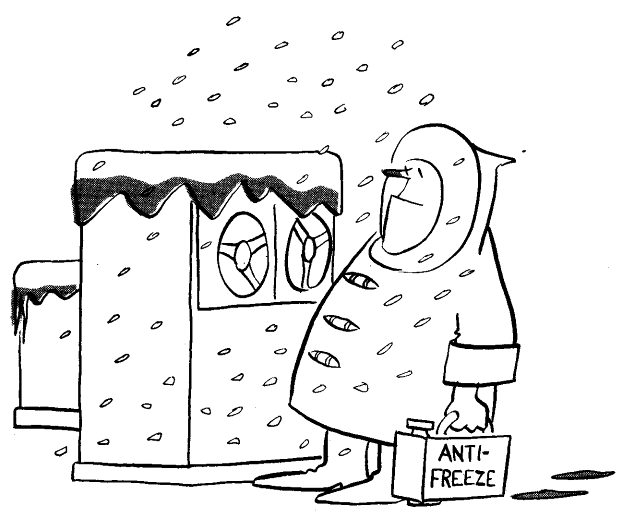
Our thinking, however, was surely not backward. I was assigned for a time to the construction effort of a very pre-FORTRAN compiler which we called ASAIC for Airship Algebraic Interpretive Coding. As time passed we actually got ASAIC on the air, and it did succeed in reordering and compiling code from statements written in "plain" algebra.

Meanwhile, out in the great world, things were progressing at a rapid pace. Edmund C. Berkeley had published in 1948 his famous epic, *"Giant Brains, or Machines That Think,"* and he followed this by beginning regular publication in 1951 of "Computers and Automation," the first journal devoted exclusively to computing.

Berkeley was also among those instrumental in organizing the first Joint Computer Conference. This was held in Philadelphia in December, 1951, and it was actually called the "Joint AIEE-IRE Computer Conference," though Berkeley represented the ACM on the conference committee.

The keynote address was presented by W. H. MacWilliams of the Bell Telephone Laboratories, and he spoke rather more briefly than any subsequent keynoter. A multi-authored paper by J. Presper Eckert, Jr., James R. Weiner, H. Frazer Welsh, and Dr. Herbert F. Mitchell described the inner glories of Univac I, which had passed its formal acceptance tests in March of 1951 at the Bureau of the Census. The best IBM seemed able to counter with was a paper entitled "The IBM Card-Programmed Calculator," by John W. Sheldon and Liston Tatum. There were papers on various extinct, one-of-a-kind machines, including the ORDVAC, the EDSAC, the SEAC and Whirlwind I. The papers on The Burroughs Laboratory Computer, by G. C. Hoberg of the Burroughs Adding Machine Co., and on the ERA 1101 by Frank C. Mullaney of Engineering Research Associates, Inc., gave promise that new competition might enter the fray.

The conference was heavily dominated by hardware papers, and the final comment of the proceedings, stem-



ming from the discussion group on "Universal Instruction Codes," seems to summarize the then-current view. This comment was: ". . . the problems of programming at this point appear as yet mostly unsolved."

As these great events took place in the world, and as time wore on, life at the Airship Foundry began to grate on my nerves. The perennial programming and running of other people's problems is not calculated to fulfill the artistic soul. After politely asking for a transfer to empennage engineering or to the human foibles staff, I resigned to become a high-ranking analyst with the Intelligible Assurance Society of America, the noted insurance cartel. In this move I was about to enter an entirely new world.

(Chapter II will appear in an early issue.) ■