The good old days

Ewart Matthews

It was my good fortune to be part of the earliest days of the computer age, from the birth - almost - of computers as we know them today. Of course I was not involved in the development of the initial concept, but came on the scene when computers were sufficiently advanced to be taken out of the hands of the "eggheads" and into the clutches of ordinary mortals.

Someone, somewhere took a great leap forward and after years of experimentation, finally decided to allow a normal, business person to control one of these hideously expensive devices and use it to process business data.

Installing and running a system

You could not simply plug one of these behemoths in and boot in a program. Programs as such did not exist in the commercial world. Office and accounting work was done manually by clerks. At the start, no one knew how to make a machine do all the individual tasks that were being done manually. So, not surprisingly, many mistakes and disasters accompanied the learning process. Nor were there any compilers to put things together. You worked out what you wanted the machine to do in a rough kind of way using hand drawn charts, and then transposed this into a program, written in machine language. You typed the program into a typewriter that also punched paper tape. The paper tape was then fed into the computer on using a "high speed" tape reader. This was followed by an agonising process of trial and error, using sample data, before live data was used to produce a real result.

There was nothing even remotely like DOS. Machine language was just that, long lines of numbers and letters, each carrying some message, something like the switches in a DOS command line, and just as mysterious to a beginner.

Things we take for granted - compilers and other programmer's aids - came years later, when magnetic tape appeared. Display screens were in the future, too (television was experimental, but no one had seen one at that stage). Magnetic tapes, magnetic stripes on cards, disks, all common now, were just dreams for the future.

Input/Output

The only input/output device available to the programmer was a console typewriter and all contact with a running program was through the typewriter, at typewriter speeds.

Once a program had been hammered out on the drawing board and punched into paper tape, it could be entered each time it was needed by using the typewriter keyboard again to enter a little, starter-program that instructed the computer to "read tape into location X." Of course the first memory location this tape program sought out was at "X." Because this was likened to coaxing a fool to "lift himself by his bootstraps," these little "get started programs" (similar to programs like "install" that we often use today) became known as "bootstraps."

The bootstrap or "boot program" would cause the typewriter to type "enter date". You would comply by typing in today's date. This was done one finger at a time in the early days! The processor would then type out "today is (say) Thursday" which put the ball back in your court to confirm the date was, in fact, correct.

One good thing about this early system was that you had a typed record of everything the operator had done. Wouldn't that be useful nowadays?

Once a program was in the computer's memory (more of this later), it could be output only by using a free standing peripheral which contained a paper tape punch (operating at 110 characters per second). Initially, the tape output spewed the loose punched tape into a bin. Later a winder was included which wound the output tape onto a spool. Of course it was inside out and it had to be rewound on a hand cranked winder before it could be reread into the computer's memory this time, using a high-speed paper tape reader (operating at 1500 characters per second!). The paper tape reader operated at such high speed, a winder for it was never developed, and the tape always emerged in a great messy heap into a large bin and had to be rewound later by hand.

Memories

My company (known then as NCR) had a computer running in Sydney which, to my knowledge, used a unique type of memory.

Each memory location (now known as a byte), consisted of a five-inch diameter loop of spring steel wire, firmly held at each end and suspended in a solid block of insulating material. (For readers who don't remember, or never learnt the old imperial system, 5 inches is 12.7 cm.) A coil surrounded the wire at one end. When the coil was magnetised by a pulse of current, it caused a physical vibration in the wire, much as a bell rings when struck. The vibration travelled round the wire and arrived at the other end in due course (some milliseconds later) where a second coil was waiting. The physical vibration induced a small current (a spike) in the coil that, when suitably reshaped (squared) and amplified, fed back into the first coil and produced another vibration to begin its path round the coil.

If the vibrations were timed carefully, they could be fed into the wire consecutively, following each other in a pulse train until the wire "filled up". This constituted one "byte" of information - 12 pulses (with gaps, or no pulses, for the zero bits). This could cycle through the loop, ad infinitum if required, and so one byte of memory was established. There were thousands of these memory locations, I forget how many exactly, but each unit was about the size of a VCR tape cartridge. Several of these were contained in a cabinet the size of a locker-room cabinet and the cabinets stood in long rows. As the vacuum tubes aged with use, their output deteriorated. A daily exchange schedule was necessary to ensure all valves were operating at peak output at all times.

All the amplifiers involved were the old thermionic type of tube, with their glowing cathode heaters pouring out heat. They required massive air-conditioning systems simply to pipe this wasted heat away. Of course for any respectable sized memory you needed an entire floor of space to hold the many cabinets, each of which was filled with these glowing tubes pulsing with heat.

An early hard disk

In one respect this T-model computer was well ahead of its time. It had an enormous hard disk. Literally! It was about a metre across and it was mounted vertically. To operate accurately, the disk had to be isolated from any vibrations, so it was mounted on a concrete platform. I cannot remember capacities or information exchange rates for this disk, but I do remember that it served the system well. It was in fact error free, until after some years of service the bearings were changed. It never worked successfully again. What's that old saying, "If it ain't broke, don't fix it?"

This early disk was so successful that I cannot account for the long delay in developing today's hard disk. I don't know why it took so long to reinvent the wheel, as it were. But when I was sent to the US for training in 1960 we used computers with memories made of ferrite cores. In this system each "byte" consisted of a string of 13 tiny doughnut-shaped rings about the size of the head of a pin. The cores strung in lines of 13 (we used 12 bits and parity) made up one byte of memory, although we called them "slabs" as sort of shorthand for "syllable" of memory.

Was this system an improvement over the earlier one? Yes. It was faster and smaller, a trend had started!

The cores had three coils wound round them, a "write" coil, a "read" coil, and a "sense" coil. When current was passed through the "write" coil, the core was magnetised in a certain direction and represented a "one" bit. During this time the sensing circuits would be disabled and cores containing zeroes remained inert.

Once data was entered it was more or less permanent, at least until the program changed it or all zeros were written to "clear memory."

To read the memory location, current would be sent in the opposite electrical direction through the "read" coil reversing the magnetising effect in cores that contained "one" bits and producing "blips" in the now enabled sense coils.

The resulting "blip" caused a circuit called a "flip-flop" to flip - or change state. That memory bit then resided in a register as part of a binary number or letter - depending on whether you had asked for alpha data or numeric data. Each 13-bit string of cores was read out at the same moment, timed by an internal clock.

Each memory location contained 12 data bits, which could represent either three 4-bit characters ("numerics") or two 6-bit characters ("alpha"). You can still see this pattern in an ASCII code chart, the two higher bits being "zone bits" to give the corresponding four levels of the ASCII system.

You could make a mistake and read "numeric" data as "alpha" or vice versa. When you did the output was called "nonsense English" and it was up to you to sort out the mess.

In this computer, the circuits were arranged on large removable boards in groups of 1000 memory locations, called planes. Ten planes fitted into a cabinet about the size of a domestic refrigerator. Our system had four planes, containing 40,000 memory locations - 40 KB of RAM!

That was the only memory in that computer, there was no hard disk. Data was processed record by record, in via punched paper tape or punched cards, out (after sorting, etc.) via punched paper tape, or as data sent to a printer.

With some understanding of the complexity of processing data on these early machines - especially when compared to modern equipment and software - you can get a feel for how clumsy and inefficient hand-accounting methods must have been. It's hard to believe that the cumbersome process I have described was an improvement on the manual methods! If you still need convincing, I can only point to the alacrity with which most firms switched from hand to electronic data processing.

The central processor

The whole computer system revolved around a crystal oscillator clock which beat out 6-microsecond timing pulses. These were tapped off round the processor in one-tenth microsecond divisions so that events could be manipulated to occur at exactly the right instant anywhere in the computer. Each cycle within the computer took 6 microseconds!

With systems nowadays operating at ever increasing speeds this seems awfully slow. But because input/output via paper tape and punched cards was also slow, the computer's speed was actually far beyond the needs of the day. We never had the experience of waiting for the computer to complete a task, in contrast with our experience with today's high speed computers. (In fairness I admit that we did wait for those long, long reels of paper tape and mountains of punched cards to feed through a reader.)

These systems were very sensitive to electrical interference. Power was produced from the mains by a large motor/alternator set that output 110 volts in three phases and served as an effective flywheel that completely blocked the system from line interference.

The circuits were mounted on printed circuit boards using discrete components that had to be hand assembled in the factory. The boards could be pulled out when we needed to troubleshoot faults from a bad diode or a resistor "gone high." There were four "flip-flops" to a board and several cabinets crammed with these racks of boards. In general the CPU was reliable, rarely giving us trouble, for which we were duly thankful.

Where did the data come from?

Each cash register or accounting machine in the offices and stores was equipped with a paper tape punch, mounted under the counter. At the end of the day the punched tapes were gathered and sent to a "data processing centre" where computer staff worked through the night, feeding the tapes through and processing the data as it went in. Even then the computer was fast enough to sort and file data as the data was being entered into the system via the paper tapes.

These punched paper tapes carried data in the form of seven holes (six bits and parity) along with a smaller, central sprocket hole. The sprocket hole was engaged by a toothed wheel for steady feeding and to ensure no slippage occurred while the tape was being punched.

When the tape was in the reader the small sprocket hole acted as a clock generator. Being smaller than the data holes, it ensured that all the data holes had been "seen" by their photo cells and that adequate time had passed for the signals to steady before the clock hole registered and the character was sent to the processor.

Punched cards were input using a 450 card-a-minute reader. Each card contained 80 columns of information, each consisting of 12 bit positions. That is to say, each row represented a complete memory location or byte.

When the card was fed out - during reading - the trailing edge of the card uncovered a series of 80 holes in the table of the card reader and exposed the holes to light. The holes were uncovered one by one and so provided a clock pulse (in the same general way as was done by the sprocket hole in the paper tape reader) to time the sending of a row of information to the processor. Once the "input run" was finished, most of sorting and processing work was also done and printing could continue.

Output

Today's "high speed" printers have not advanced a great deal from the early days, at least from what I can determine. Our old battler could print 1500 lines a minute of alpha/numeric data and 3000 lines a minute of all numeric characters. Each line was 150 characters wide, so the speed with which the continuous stationery moved through the printer was already close to the maximum possible. When that printer was producing invoices, the paper poured through at great speed.

The typeline contained a complete set of alpha characters and had room for two sets of numerics in each of the 150 positions. This meant that data that was all numeric could be printed twice per revolution of the typeline. A welcome innovation that was to come later, was a mechanical output feeder and stacker for the high speed, all-numeric work.

Of course moving this much paper and cardboard at high speeds generated plenty of static electricity, which could be disastrous because it caused memory errors. So it was important to control the moisture content of the paper tapes and cards, another reason these large systems needed such extensive air-conditioning systems.

Giving the input materials time to equilibrate with the conditions in the computer room was another precaution we took to protect against shocks. Storing the tapes and cards in the computer room environment for several hours gave the paper and cardboard time to settle at room temperature and humidity. Nevertheless I had many electric shocks handling a bin (on rubber wheels and insulated) which picked up surprisingly large amounts of charge after a long input sessions, so we devised steel shorting fingers on the cabinets to short these tape bins to ground.

We technicians worked pretty hard to keep everything running smoothly. The systems ran 24-hours-a-day, seven-days-a-week and we were only allowed to have the system for one hour a day for maintenance. This meant we spent long hours out in the workshop, changing boards and trouble shooting problems.

Gradually modern equipment began to replace the original machines. Computers became faster, smaller and more reliable. But I have fond memories of those early days and of a stimulating and satisfying job which I loved doing.

Reprinted from the April 1996 issue of PC Update, the magazine of Melbourne PC User Group, Australia