Digital Engineering is More Fun

“Core dump? What does that mean?” our teenaged Unix-wanna-be inquired. I looked up, and thought (not for the first time) that young employees sure keep the mad dash of technological change in perspective. Their questions often require more thought and introspection than those of their elders. We generally have a couple of youngsters working here, both to give them the chance to see what engineering is all about, and to invest in those sparkles of brilliance that only rarely appear.

I pulled out one of my holy relics - a 3 pound, 13,000 bit core array acquired from a surplus shop in 1971. A few days after high school graduation I hitchhiked with a pal to Boston (those were kinder, gentler days) to find treasures in the disorganized depths of a surplus shop.

Was it our long hair? Maybe the fact that we were warned three times to get off the New Jersey Turnpike had something to do with it. Somehow Gary and I found ourselves in a New Jersey jail cell, busted for hitching. The police, expecting to find a stash of drugs in our backpacks, were surprised to discover instead my 13,000 bits of core.

“What’s this?” the chief growled. I timidly tried to convince him it was computer memory. These were the days when computers cost millions and were tended by an army of white robed technicians, not hitchhiking hippie-freaks. All of the cops looked dubious, but could find nothing to dispute my story. They eventually let us go, me still clutching the memory.

A few years later I experienced an eerie echo of this incident, when I lived in a VW microbus. Coming back from Canada into a remote Maine town the local constabulary, sure I was running contraband, stripped the van. They found a 6501 - the first, low cost microprocessor chip. MOS Technology amazed the electronics world when they released this part – the predecessor of the 6502 of Apple II fame – for only $20. I just had to have one, though I just tossed the part in the glove compartment. Here too, the officers were unaware of the pending microprocessor revolution, and were equally disbelieving about my story about “a computer on a chip.” Sure, kid.

Coming back to the present, I showed my employee the core matrix. You know how teenagers are - she didn’t believe a word I said. “Come on, that big thing only stores 13k bits? It’s bigger than my whole computer,” she noted, disdainfully, as if I were a stone-age man showing off my cave painting technology to a Corel whiz.

Looking closer I realized that 10 of the cores covered an area about the same as a 64 Mbit DRAM die. How things have changed!

My college had a Univac 1108, a $10 million mainframe that serviced all of the university’s students. We upgraded the machine, adding 256k words (36 bits each) of additional core memory, at a cost of $500,000. The core featured a 750 nsec cycle time and was the size of several refrigerators. For months, until Univac’s engineers got the bugs out, leaning on the box was sure to cause bit dropouts. This almost always crashed the system and drove users into an uproar, especially during finals week when projects were due. The machine serviced some 500 users simultaneously, yet had only 768k words of memory, much less than in one of today’s laptops. A six foot long drum memory stored something like 50 megabytes. The 1108’s claim to fame was instruction execution in under a microsecond, a claim that pales next to the Pentium’s few tens of nanoseconds… for a price that approaches zero dollars.

No one dreamed of personal computers. Sure, a few stories circulated about the surplus deal of the century, where a clever person picked up an old 7094 for a pittance. Few could afford the electricity, let alone the astronomical air conditioning costs required for running such a monster.

IBM’s 7094 was a wildly successful mainframe. My school had one, for a while, till replaced by the Univac. Core was so expensive that the entire system, the computer that ran a 40,000 student campus, had but a pathetic 64k - with no disk storage. Programs wrote temporary information to tape, keeping a room full of drives whirring constantly. There was no security to speak of. One of our great delights was running a program that searched for teachers’ programs on the input tape, always with the hope of finding grades that could be, ah, "debugged". Eventually the university wised up and bought an additional Univac with no student accounts or dial-in lines, which ran all of the grading and billing software.

Throughout this period core was the only random access read/write memory in common use. It wasn't till the very late 60s that even the smallest MOS memory chips became available.

Each core is a ferrite bead, perhaps the size of a small "o" on this page. Four wires run through the center of each core, four wires tediously strung, by hand, by Asian women who no doubt worked for a pittance.

Cores are tiny magnets, each remembering just one bit of information. The trick is to flip the magnetic field of the cores -- one direction is a "one"; the opposite field indicates a "zero".

As we know from basic electromagnetics, a changing voltage creates a magnetic field, just as a change in a magnetic field induces a voltage. The wires running inside of the ferrite beads create the fields that flip the direction of magnetization, writing a zero or a one. They also sense the magnetic field so the computer can read the stored data.

Two of the wires organize the core into an X-Y matrix. The core plane is an array of vertical and horizontal wires with a bead at each intersecting node. Run 50% of the power needed to flip a bit down each wire - at the intersection there's all that's needed to flip just that one bit. What a simple addressing scheme! As the bit changes state, it induces a positive or negative pulse in a third wire that runs through all of the cores in a plane. Sensitive amplifiers convert the positive or negative signal to a corresponding zero or one.

Since the amplifiers detect nothing unless the core changes state, reads are destructive. You've got to toggle the bit, and then write the data back in, on each and every read cycle. It sounds terribly primitive till one thinks about the awful things we do to keep modern DRAM alive.

Before microprocessors quite caught on, the instrumentation company where I worked embedded Data General Nova minicomputers into products. The Nova used core arranged in a 32k x 16 array. The memory is non-volitle, remembering even when there's no power applied.

We regularly left the Nova's boot loader in a small section of core. My fingers are still callused from flipping those toggle switches tens of thousands of times, jamming the binary boot loader into core each time a program crashed so badly it overwrote these instructions.

For some reason these Nova memories suffered a variety of ills. Core was expensive - around $2000 for 32k words, at lot of money in 1974 dollars. A local shop repaired damaged memory, somehow restringing cores as needed, and tuning the sense amplifiers and drive electronics.

As we worked through these reliability issues, my boss - who was the best digital designer I've ever met - told how some military and space projects actually employed core as logic devices. In a former job he designed systems composed of strands of core strung together in odd patterns to create computational elements. I remember being just as incredulous at his stories as my teenaged employee is at mine.

Yesterday I read of a new ferromagnetic technology NEC is coupling to DRAMs for increased memory density. Is magnetic memory soon to make a comeback in a new guise?

Just as core memory itself is an anachronism, so is the concept of the core dump. Who has the energy to examine a binary memory image anymore? Surely, if this is to be a useful debugging methodology, at least some sort of symbolic information should be included in the dump.

Unfortunately, Unix continues to litter our drives with sendmail.core, httpd.core, and other binary images, each a snapshot of the death throes of a program. I'll set my teenaged employee looking for a way to disable the dumps. Perhaps twenty years on, she'll regale a new generation with her stories of the bad old days of computers, before brain implants supplanted keyboards, when people were not yet integrated into the global data processing network.

The Ganssle Group
PO Box 38346, Baltimore, MD 21231