

## Evolutionary Dead Ends

(Or How Good Ideas that Successfully solve problems are not always the way of the future )

### Abstract

There were five things about the AGC which, like an ugly child that only a parent can love, stand out in retrospect.

- Core-rope (aka ROM) program memory.
- Direct Memory Access to specialized registers as a way of doing I/O.
- One's complement number system, instead of two's complement.
- Verbs and Nouns, a Prehistoric Command Line interface.
- A single IC logic element.

Each of these design features solved some problem, each had consequences we did not foresee, and each was left behind in standard computer architecture.

### Introduction

As always, there is a beginning before the beginning. I joined Eldon Hall's group at MIT's Instrumentation Lab in 1957 or 58, at a time when Eldon was consumed with the Polaris missile GNC computer. I wasn't much use to him for that job, being a fresh PhD without much practical experience, and from Harvard's Computation Laboratory yet, so I wound up joining forces with Hal Laning, Eldon's boss at the time, in designing a GNC computer that might do the job in a proposed Mars probe. That computer had to be lighter than anything available at the time, and, more important, had to use many fewer watt-hours for the mission.

We thought we could take advantage of several things, some of which turned out to be pertinent for Apollo:

- The computer need not be working all the time. In fact, for the Mars probe, it only had to work a small percentage of the time.
- Core-logic, really transformer coupled logic, could preserve the computer's states without power.
- Interrupts could present the IMU increments of angles and velocity in internal registers, without "conscious" program intervention, i.e., the hardware caused the current, correct values to be there.

We designed and built a prototype with a 12 bit word length, a Core-Rope memory, and DMA interrupts. The Mars probe never happened, but Apollo did.

## The AGC Core-Rope

I first learned about the Core-Rope from an Electronic News item that described a Read-only Memory that used a single nonlinear magnetic core transformer per word. The Ones were wires that went through the core, while Zeroes were wires that bypassed the core. The cores were molly-permalloy ribbons wound on a ceramic bobbin. Because they were non-linear, with a well defined hysteresis loop, selecting the core could be done by inhibiting all cores but one, and then pulsing all the cores simultaneously. The inhibited cores were biased so that very little signal coupled to the bit windings, while the uninhibited core switched through a relatively large change in flux. The advantage of this scheme was that  $n$  pairs of bias lines allowed inhibiting  $2^n$  cores. The only alternative to this sort of memory would have been a coincident core memory, which would have been much larger, and which required restoring what was read out, and hence potentially susceptible to losing information. As it was, the final AGC configuration had some 72K bytes of ROM, and 2K bytes of coincident core RAM, to use modern language. The advantages were clear: low (relatively) power, no power on standby, no loss of bits because of electrical accidents. One disadvantage was that it was relatively hard to build, and even worse, in the eyes of the software developers (at least initially), was that you had to have the final program before you built it. In fact, the program had to be frozen several months before it was needed for final checkout in the Spacecraft.<sup>1</sup>

Needless to say, ROM based computers are not mainstream architecture these days, not even for Operating System kernels. However, for Apollo the disadvantages turned out to be major advantages in the long run, because, once the S/W developers came to grips with the fact that they could not walk up to the launch pad at the last minute to load their latest version, they were forced to freeze everything early, which gave the code reviewers and testers stable code for enough time to do a good job.

## Direct Memory Access

I remember that the subject of how to get the data from the IMU, angles and accelerometers, into the AGC, was a hot topic for a while. A conventional general purpose computer, which we thought was what we were designing, would invoke subroutines which would in turn use some sort of special hardware. Hal Laning did not like the idea of interrupting the flow of computation to go get the necessary data, because of the many memory cycles it would require, so he and I, at the time of the Mars computer, invented (with many others, it turned out) and got a patent for, what today is called Direct Memory Access.

We identified certain registers from the Erasable (RAM) Memory (which was a regular coincident core memory, in the AGC) as the places to go look when you wanted to refer to a specific angle or velocity. The increments of angle and of velocity were pulses that came into the AGC via special circuits that:

- Saved the state of the Accumulator

---

<sup>1</sup> But there was at least one occasion when a spacecraft problem in flight resulted in a last minute, emergency program transmitted verbally, manually entered into Erasable memory and executed. Best laid plans, etc.

- Incremented (or decremented) the value in specific RAM addresses.
- Restored the state of the Accumulator.

The difference between this scheme and a regular interrupt is that in the latter you have to save the state of the program counter and any other CPU registers, then invoke an interrupting subroutine, and later restore the CPU to its original state. In the AGC scheme, the regular program flow did not have to concern itself with how the quantities got to those registers, and their latest value could be counted on (no pun intended) to be there no matter what the program had been doing.

This scheme was really successful, so much so that IBM's Federal Systems division, at the time of some contention between it and MIT's Instrumentation Laboratory, referred to the scheme as "Cycle Stealing." But this scheme, too, did not survive into the general computer world, possibly because general purpose computers don't (usually) have the bulk of their I/O in the form of pulse increments and decrements.

### **One's Complement number system**

The original Mars computer did not even have a proper adder. Sum and Carries were kept in separate registers, and addition was a subroutine that kept repeating until there were no carries left. We chose a One's Complement numbering system because in it the negative of a number is a simple bit-by-bit inversion, while in a Two's Complement scheme you first do a bit-by-bit inversion, and then insert a carry into the least significant bit position, which would require lots of add cycles for going from  $-1$  to  $+1$ , and vice-versa. One thing to note, because it comes into the story later, is that a One's Complement system has two representations of zero: all 1's, and all 0's.

Now, the AGC did have an adder, but for reasons I cannot remember we carried the 1's complement scheme over. Through Block I, and into Block II, everything seemed fine as far as the arithmetic of the AGC was concerned<sup>2</sup>.

A prototype AGC found its way to an integration lab, where it got married with an IMU, two DSKYs, optics, etc, and tests began. And shortly thereafter Al Hopkins and I began to get reports that "the AGC was dropping pulses," which, like all hardware designers, we attributed to "operator error." Since saying this affects systems integrators as a red flag does a bull, we soon got chapter and verse on how to reproduce the problem. It seems that the increments of angle that came from the IMU were one short every day, i.e., for every rotation of the earth. The IMU designers, knowing that the AGC word length was 15 bits, had divided the circle into  $2^{15}$  parts. But no one had told them (it had never occurred to me) that, since the AGC uses 1's complements, there were only  $2^{15} - 1$  unique numbers. Zero could be either all zeros or all ones. Hence, every rotation of the earth the AGC adder would happily increment from one version of zero to another, leaving the register contents unchanged.

---

<sup>2</sup> At one point, though, someone found that there were 13 values for which division came out wrong. Nothing to do with 1's complement, though.

My proposal that the IMU reducing gears be redesigned to account for that one part in 32,768 was not well received: for one thing, that number has 15 factors, all 2, so there are lots of choices for gear ratios, while 32,767 has only 7, 31 and 157 as factors. My comment that it could have been worse, that  $2^{15}-1$  could have been a Mersenne prime, went unappreciated.

Herb Thaler did his magic, and by changing only the wire-wrap connections, fixed the problem.

I don't know of any other computer that uses 1's complements<sup>3</sup>.

### **Verbs and Nouns, a Prehistoric Command Line interface.**

At one point, while the overall design of the AGC was reasonably well established, the human interface was not. All that was firmly established was the physical dimensions of the two DSKYs, as they would eventually be called. Jim Nevins's group was in charge of "human factors," which briefly considered (then) far out technologies such as CRTs, but the power consumption constraints soon resulted in a consensus that Liquid Crystal Displays, and latching relays, were the way to go. In the meantime, we, in Eldon's group, had a computer with no way of showing visiting dignitaries and firemen that it was doing something useful, so we were in a waiting mode while the human interface got settled.

One day, while threading my way into work by way of Cambridge's back streets, it occurred to me that the sort of dialog between astronaut and AGC could fit into a rudimentary sentence structure, such as "Display IMU Angles," or "Display Time," or "Fire Rocket," or "Align IMU." With Al Hopkins and Herb Thaler, we soon settled on a design we could use as an in-house demonstration DSKY, while waiting for the "official" design. Our DSKY would have a two-digit numeric verb, a two-digit numeric noun, and three five-digit display registers, for the three vector components of velocity or anything else that might fit. By this time important items such as the size of buttons (so astronauts could operate them with gloves on), and luminosity were decided, so we were able to build a prototype that fit the physical and human requirements.

As we proudly demonstrated the AGC displaying time, or some fake velocities, to visiting dignitaries, we soon began to sense a pattern to their questions. They started with: "this verb and noun, is it going to stay?" We explained that what they were seeing was our internal, temporary display, and that the official one would be coming soon. But, as the same question kept coming up, we began to ask what the objection was to VERB-NOUN.

The objections were interesting:

- It's not scientific.
- It's not military.
- It's not serious.

---

<sup>3</sup> Back in the 60's I reviewed a paper that proposed a **negative** radix computer. I.e., the lsb would be worth -1, the next bit 2, the next one -4, the next one 8, etc. It was supposed to have some esoteric advantage in multiplication.

- It's not dignified.

We would also ask for alternatives. "ACTION" in place of "VERB" might work, given the constraint that the engraved title letters had to be visible through the astronaut's visor, and hence had to be of a certain size. But then, replacing "NOUN" with "OBJECT OF THE ACTION" would not fit.

The final DSKY was eventually pretty much what we had designed as a throwaway, with the addition of a two-digit MODE display. And the final objection to VERB-NOUN came from a senior member of the lab, who had never actually been up to see the computer he had helped program. He, too, thought it was not dignified enough. So, being experts by this time, we baited the man for alternatives he could not come up with. As we pushed and pushed, and he got more and more defensive, he finally came up with the fundamental, final reason why VERB-NOUN had to go:

- Because the astronauts would not understand it.

Years later, the Computer Museum opened and chose the AGC as the subject for its dedication. At the ceremony, Dave Scott gave a speech describing his experiences, and at one point he said (as best my memory tells):

"... and I don't know who thought up that VERB and NOUN scheme, but we liked it a lot, for it made matters very clear to us as to what we were seeing, or had to do."

Minimalist command line interfaces are to be found no more in computers, but I still get a good feeling remembering Dave Scott's words.

### **Integrated Circuits – One Logic Type**

Al Hopkins and I were heavily invested in Core-Transistor logic, which we had developed for that Mars computer. Eldon Hall had his doubts about it, and started Herb Thaler working on some newfangled Integrated Circuits. At one point, with a breadboard AGC well under way, Eldon informed us that, despite the brilliance of our arguments, the work to be thrown away, the program delays, etc, it was going to be a single 3 input NOR gate, and not Core-Transistor Logic. By comparison, the Minuteman missile was designed with a dozen or more different circuits. Al Hopkins and I, kicking and screaming, finally agreed to the change, and now with Herb Thaler joining us, redesigned the prototype to accommodate NOR gates in a matter of weeks.

The discipline of using a single type, and to force this single circuit element into the CDU and the GSE appears to be another evolutionary dead end, but one that Al, Herb and I now think was essential to the success of the AGC – and the Apollo program. It is now conventional wisdom that quality, reliability and lower cost increase with volume, but it was not so then.