My analog to digital conversion

Les Earnest (les at cs.stanford.edu)

Abstract
In the 1950s I helped design the SAGE air defense system, which was a technological marvel that included the first computer network. Unfortunately it didn't actually work but gave rise to a military-industrial-political complex that has bilked U.S. taxpayers out of many billions of dollars and is still going strong. Along the way I managed to get permission to use nuclear warheads based on an analysis that, based on 50 years of hindsight, was somewhat defective.

Introduction
I was born in San Diego, California, in 1930 in a largely analog world. There was no television but lots of AM radio and "computers" were people, mostly women. My father was a self-taught electronic (mostly radio) engineer who had gotten into it as a teenager and went on to become general manager of a manufacturing plant, then a municipal administrator, where he initiated the development of San Diego's rather elegant Mission Bay Park. My mother had been a popular actress in local theatrical productions but turned down an invitation to go to Hollywood and chose instead to become a teacher and mother of two while earning a Ph.D. in her "spare time," then become a college professor.

Quick draw slipstick
I somehow got admitted to Caltech as an undergraduate and followed the nerd norm there of wearing a twelve inch slide rule in a leather scabbard strapped to my waist, so that I could make a quick draw to begin computing at any time. It was a Keuffel & Esser LOG LOG DUPLEX DECITRIG and I used it in all classes and for homework.

I particularly enjoyed my freshman chemistry lectures, taught by Linus Pauling, but was puzzled by some of the computations he made in class with the assistance of a 6 inch slide rule. The strange thing was that he wrote down answers to 5 or 6 significant digits even though such a slide rule is good to no more than three digits. There was no such thing as a pocket calculator in that era, so we couldn't check him on the spot, but when I wrote down the numbers and later did the calculations on paper his answers turned out to be correct to the last digit!

My classmates and I puzzled over this for some time, wondering if he knew some magical way to get more digits out of a slide rule. Finally, someone asked him in class how he did it. He remarked that he had worked out the problems in advance on paper and used the slide rule as a mnemonic device – it gave him the leading digits and he remembered the rest. I later learned that Professor Pauling had a photographic memory, which made this relatively easy for him.

Following my freshman year I landed a great summer job, working with several other students as "guinea pigs" at the Naval Electronics Laboratory in San Diego. Our first assignment was to listen to digitized speech samples in experiments designed to determine the sampling rates and resolution needed to reproduce intelligible speech. This very far-sighted experiment was initiated and conducted by JCR Licklider, who was a most enjoyable person to work for. As luck would have it, Licklider later played a prominent role in initiating the development of the Internet and we reconnected a few more times during the next thirty years.

That experience and others got me to thinking about the contrast between analog and digital computation and I decided that I preferred digital. I replaced my slide rule with an abacus purchased at a local Chinese market and hung it from a strap on my belt. It was the only kind of portable digital calculator then available and had the beneficial side effect of unnerving my classmates during exams as I clicked the beads while they silently slid their slide rules. I even learned how to compute square roots on the abacus, which was not a widely known skill.

I had initially decided to be a chemist, having been inspired by Linus Pauling, but eventually concluded that I wasn't cut out for it. In the meantime I had gotten interested in electronics, fiddled with radios, and built a couple of Theremins (electronic musical saws), so I switched to electrical engineering, expecting to get training in electronic circuit design. Alas, it turned out that all the Caltech EE professors at that time were power men who didn't understand electronics. I eventually did get one course in the design of vacuum tube circuits, taught by a physicist. However I learned nothing about transistors. They had been invented when I was a senior in high school but apparently nobody yet understood how they worked.

Feeling a draft

Approaching graduation from Caltech in 1953 I faced the likelihood of being drafted into the U.S. Army for service in the Korean War. However with a bit of research I found a unique Navy program. They required a three year commitment compared with two for the Army but made you a Restricted Line Officer provided that you had both an engineering degree and poor eyesight – perfect!

I was accepted and was sent to Officer Candidate School in Newport, Rhode Island, for the summer followed by another school in Jacksonville, Florida, for the winter – a pretty nice start. During my stay in Newport they tried to get us to volunteer one weekend to help conduct the wedding of the junior senator from Massachusetts, John F. Kennedy, to Jacqueline Bovier. Being a Southern California beach boy I chose instead to go to the beach and never regretted it.

Unfortunately the following summer I was sent to the Navy Aviation Electronics School in Memphis, Tennessee, which was not a happy place for a beach boy. I viewed that area as the armpit of the U.S. and, to make matters worse, even though I was a freshly minted electrical engineer, the electronics course there began at the bottom with Ohms Law. I started reading novels in class but did pay enough attention to learn the difference between U.S. Navy electronics and that of the rest of the world. Specifically in the Navy electrical current ran from – to + rather than from + to –.

I eventually discovered that I could shorten my stay in Tennessee by taking exams without doing the course work and soon qualified for my next assignment, which turned out to be at the Naval Air Development Center in Johnsville, Pennsylvania, a suburb of Philadelphia.

Typhoon and the CPC

I found that the Aeronautical Computer Lab there had the largest and fastest electronic analog computer in the world, called Typhoon. It had been built by RCA and was used for aircraft and missile flight simulations. They also had a small digital computer, called the IBM Card Programmed Calculator (CPC), which had originally been created out of accounting machines by engineers at Northrop Aviation.

The CPC was used to do check solutions for Typhoon, given that there were a lot of ways in which mistakes could be made in programming an analog computer. I was offered a choice of what to work on and, based on my natural preference for digital devices, chose the CPC. I learned that doing flight simulation typically involved generating numerical solutions to something like twenty-seven simultaneous differential equations, which was a challenge inasmuch as the CPC had only about forty words of electromechanical storage. However programs were stored on punched cards that were fed through the card reader and there was no limit on their size.

The programmers there were all mathematicians and I learned that they viewed the CPC as a black box whose internal workings didn't interest them. However after learning to program I started looking inside and discovered that the designers of this system had done a sloppy job in some respects. I rewired the plugboards so as to improve its performance by providing an additional data path and faster computation on certain transcendental functions.

As I approached completion of my Navy obligation in 1956 I was pleased to discover that computer programmers were in demand – there were probably only a few hundred of us in the whole world at that time and a number of corporations were looking for more. Through my Navy contacts I had been hearing about another windy project that had been funded by the Navy at MIT called Whirlwind. It too was originally planned to be an analog computer for flight simulation but had later been converted to digital, then abandoned by the Navy. Nevertheless their technology was quite interesting – indeed, the head of that project, Jay Forester, had invented core memory, which revolutionized digital computers in that era.

Whirlwind turned out to be a solution looking for a problem and MIT eventually talked the Air Force into making it the starting point for an air defense project called SAGE (Semi-Automatic Ground Environment). Given that their computers needed to provide guidance to manned interceptors and missiles, which is what I had been working on for several years, and that I also knew how to program I was recruited to MIT Lincoln Lab to work on weapons integration for SAGE.

SAGE like Forrest Gump

SAGE ostensibly was developed to counter the threat of a manned bomber attack by the Soviet Union, but it was also served to counter the political threat of a competing air defense system called Nike that was being developed by the Army.

SAGE was the first large real time computer system. "Large" was certainly the operative term--it had a duplexed vacuum tube computer that covered an area about the size of a football field and a comparably sized air conditioning system to take away the enormous heat load. SAGE collected digitized radar information from multiple sites and used it to track aircraft and guide interceptors. It was designed to work initially with manned interceptors such as the F-102, F-104, and F-106 using a packet radio system to transmit guidance commands to these aircraft. It was later modified to work with the BOMARC ground-to-air missile.

In late 1956 when I began work at MIT Lincoln Laboratory I was assigned to share an office with Paul Sinesi. When I asked what he did, he replied with a proper Boston accent, "I work on rada' dater." I knew that SAGE was supposed to automatically track aircraft movements based on radar data, so I asked "What do you do about electronic countermeasures?" "We don't do that" he said.

Given that I had been trained as an aviation electronics officer I found that answer puzzling, since I knew that hostile bombers normally attempt to jam radars. Being new to the organization, however, I figured that there was something he wasn't telling me and that I would later learn how they dealt with this problem.

In fact, as I subsequently figured out, they didn't deal with this problem and carefully designed all tests and demonstrations to avoid it. Thus SAGE was a gigantic fraud on taxpayers in that it was a "peacetime defense system" that would malfunction under an actual attack, much like France's Maginot Line did in World War 2.

This presented me with a dilemma. I had come to MIT because it was a hotbed of computer technology, which I had chosen to pursue, but helping the military-industrial-political complex defraud the public was unethical. Nevertheless I continued to work on SAGE for several years and learned that it had other major defects. Examples:

1. Though MIT had recommended that the SAGE computer facilities be placed in hardened underground facilities, so that they would be less vulnerable to nuclear attack, that was done for only one of the 23 command centers, namely the one in Canada.
2. MIT had also recommended that the SAGE computer facilities be located remotely, away from both cities and military bases, so that they would not be bonus targets in the event of an attack. In choosing sites for the command centers, however, the leadership of the Air Defense Command apparently looked for bases with good accommodations such as golf courses and Officers Clubs. Since the Strategic Air Command (SAC) was being well funded in that Cold War era, they generally had the best accommodations. As a result, a number of SAGE facilities were placed on SAC bases where they became bonus targets.

3. Before the SAGE system was fully deployed, intercontinental ballistic missiles superseded manned bombers as the primary threat from abroad.

Thus SAGE had several things in common with the mythical Forrest Gump: it was very fast, financially successful, and incredibly stupid.

Though the designers of SAGE came to recognize its weaknesses and vulnerabilities and the Air Force should have been reluctant to build more systems of the same type, it somehow came to be regarded as the model of what the next generation of military control systems should be. Never mind that it was essentially useless as a defense system--it looked good!

The useless SAGE system was kept going for about 25 years, until 1983, at a cost of many billions of dollars because it provided a pleasant lifestyle for its Air Force operations staff, excellent places to visit for Congressmen, large incomes for their industrial constituents, and settings that looked like the right kind of place for Generals to run a war--indeed its ambience of subdued lighting and large screen displays was adopted by Hollywood for films of military command centers.

SAGE thus gave rise to a corrupt military-industrial-political establishment that has produced a string of largely useless command-control and weapons projects such as President Reagan's phony "Star Wars" defense program and the current ongoing deployment of anti-missile systems that don't work. But that is another story.

Good to go

The BOMARC missile used a rocket booster to get airborne and a ramjet to cruise at high altitude to the vicinity of its target. It then used its Doppler radar to locate the target more accurately so that it could dive at it and detonate. BOMARCs were housed in hardened structures and, when a given missile received a launch command from SAGE sent via land lines, the roof would roll back, the missile would erect, and if it had received a complete set of initial guidance commands in the meantime, it would launch in the specified direction.

It was clearly important to ensure that the electronic guidance systems in these missiles were working properly, so the Boeing engineers included a test feature that would generate a set of synthetic launch commands so that the missile electronics could be monitored for correct operation. When in test mode, of course, the normal sequence of erecting and launching the missile was suppressed.

However when we reviewed the BOMARC launch control system, one of our engineers noticed a rather serious defect: if the launch command system was tested, the missile would be in a state of readiness for launch. If the "test" switch was then returned to "operate" without individually resetting the control systems in each missile that had been tested, they would all immediately erect and launch! Needless to say, that "feature" was modified rather soon after we mentioned it to Boeing.

Which way is up?
Another problem showed up in the packet radio system. The bit allocations were carefully specified for each kind of command, such as heading, altitude and speed, and the design of the ground transmitter and airborne receivers was assigned to two different contractors, but when the system was tested it didn't work. It turned out that the specifications neglected to specify whether the high or low order bit was to be transmitted first and the two contractors made different assumptions. That too got fixed, at some expense.

Duplexed for reliability
I somehow got assigned the responsibility for leading a study to get approval to put nuclear warheads on the second-generation BOMARCs. This involved proving to a government nuclear safety board that the probability of accidentally launching a missile on any given day as a result of equipment malfunctions was less than a certain very small number and that one person couldn't do it by himself. Never mind that the idea of shooting off nuclear warheads in our own skies was insane. We did eventually get the approval but along the way we discovered a scary problem.

The SAGE system used land lines to transmit launch commands to the missile site and these lines were duplexed for reliability. Each of the two lines followed a different geographic route so that they would be less likely to be taken out by a single blast or malfunction. There was a black box at the missile site that could detect when the primary line went bad and would automatically switch to the alternate. However on examination we discovered that if both lines were bad at the same time, the system would remain connected to the alternate line and the amplifiers would then pick up and amplify whatever noise was there and interpret it as a stream of random bits.

We then did a Markov analysis to determine the expected time that it would take for a random bit stream to generate something that looked like a "fire" command for one of the missiles. We found that expected value was a little over two minutes. When such a command was received, of course, the missile would erect and prepare to launch. However, unless the missile also received a number of other commands during the launch window, it would automatically abort. Fortunately, we were able to show that getting a complete set of acceptable guidance commands within this time was extremely improbable, so this failure mode did not present a nuclear safety threat.

The official name of the first BOMARC model was IM-99A, so I wrote a report about this problem titled "Inadvertent erection of the IM-99A." While that title raised a few eyebrows, the report was destined to get even more attention than I expected. Its prediction came true a couple of weeks after it was released when both phone lines went bad at a BOMARC site in Lakehurst, New Jersey, near where the Hindenburg Zeppelin disaster had happened some years earlier. This caused a missile to suddenly erect and start the launch sequence, then abort. Needless to say, this scared hell out of the site staff and a few other people.

The Air Force was suitably impressed with our prediction and I was immediately called upon to chair a committee that had the honor of fixing the problem. The fix was rather easy: just disconnect when both lines are bad. With good engineering practice, of course, this kind of problem wouldn't occur. However, the world is an imperfect place.

Oops!

Even though the use of nuclear warheads was authorized for BOMARC B missiles, as far as I know that didn’t happen, which was just as well because under hindsight (about 50 years late), I realize that both our study group and the government safety committee overlooked the possibility that a malevolent programmer might have been able to launch a missile all by himself. There was no certainty that such a scheme would have worked inasmuch as the SAGE software was reviewed by multiple people who might have questioned any odd-looking code.

Nevertheless we should have considered that possibility. The reason we didn’t was that there was no such thing as a malevolent programmer in that era (1950s-‘60s) – we were all honest, upright, and altruistic. Thus the idea that a programmer might sneak in some code that would improperly launch a missile was inconceivable.

Later experiences on the Internet revealed other possibilities.

Moving on

After working on SAGE, I successively participated in projects supporting Air Force Intelligence, the Central Intelligence Agency and the Joint Chiefs of Staff until my concern about crooked military-industrial-political projects finally caused my ethics to boil over. I took a voluntary pay cut in order to flee to Stanford University where I had a most enjoyable time for many years as executive officer of the Stanford Artificial Intelligence Lab (SAIL).I was able to do some research, taught some classes, and eventually completed my education by doing a harrowing but successful Silicon Valley startup, but that is another story.