the Availability Digest

www.availabilitydigest.com
@availabilitydig

The PDP-8 – My Favorite Computer

April 2018

I was amused when I came across an article entitled "iPhone beaten in performance race by 1970's Apple II and other even older computers (and a mechanical calculator)."[1] The challenge was to output the numbers in the Fibonacci sequence. This sequence starts with 0 and 1, then each subsequent number is calculated by adding the two previous numbers:

0 1 1 2 3 5 8 13 21 34 ……..

Interestingly, the ratio of any two successive numbers in the Fibonacci sequence approaches a number known as *phi*, or the Golden Ratio, 1.618034.

The Fibonacci series shows up often in nature:

- The number of petals in a flower.
- The seed pods in a pine cone.
- pineapples and cauliflower.
- the way tree branches form and split.
- the logarithmic spiral seen in shells.
- the logarithmic spiral of a hurricane.

But back to the losing iPhone. Among the contenders were the following systems:

System	Relative Speed
PDP-8	16
Apple II	38
BBC Micro	70
Windows 98 PC	1477
iPhone 6	4

The iPhone is thousands of times faster than any of these other systems, so why did it perform so poorly? Simple. It cannot be programmed, so the Fibonacci series had to be spoken into it.

However, I was particularly attracted to the fact that the PDP-8 computer was included in the lineup. That has always been my favorite computer.

The Birth of the PDP-8

My love affair with the PDP-8 began long before it was born. When I was a graduate student at MIT in 1956 pursing my Master's Degree in Electrical Engineering, I worked as a Research Assistant at Lincoln

[1] *Betanews*, February, 2018.

Laboratories, which was run by MIT for the Air Force. I was assigned to the TX-0 project, an attempt to build the first transistorized computer. My immediate supervisor was a fellow named Ken Olsen. Transistors at the time cost $150 each (thousands of dollars in today's currency). If we burned one out, we had to write a full report on how it happened.

Our group was working on a multiplier. The technology of the time was to build the logic on a massive motherboard. Ken realized that this was awfully cumbersome; and he developed a technique in which transistorized logic elements like gates and flip-flops were packaged in tube-like enclosures that plugged into a wire-wrap motherboard. The motherboard backplane was wired by a programmable wire-wrap machine so that multiple units could be created easily. The result was a quick and easy way to build complex logic modules.

This technology was so effective that Ken decided to start his own company to manufacture digital systems using his packaging technique. His company was Digital Equipment Corporation. He repackaged his logic elements into small plug-in boards that he called *flip chips*. A flip chip was a single-sided 18-contact card with edge connectors.

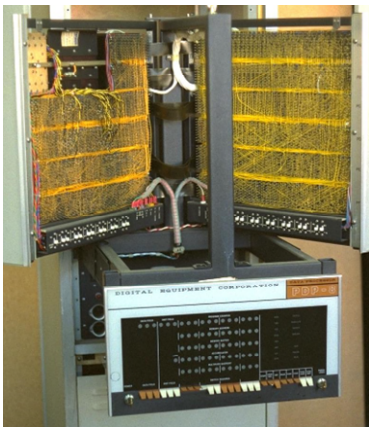
Using his flip chips, Ken went on to build a small computer, which he called the PDP-8.

The PDP-8 Architecture

The PDP-8

The origianl PDP-8 computer was sold in 1965 for a price of $18,500 ($140,000 in 2018 currency). It was about the size of a small household refrigerator and usually sat on a desktop.

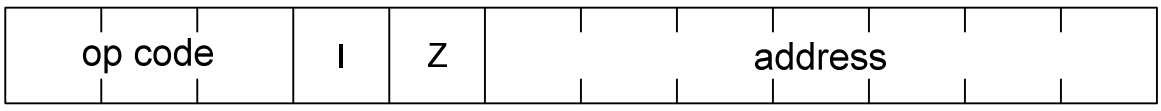
The PDP-8 was the first computer to be sold for under $20,000, making it the best-selling computer in history at that time. Over 50,000 systems were sold over its lifetime (this was long before the advent of personal computers).



The Wire-wrapped Backplane

It was also before the computer industry settled on an 8-bit byte. The PDP-8 was a 12-bit computer, 6-bit characters being the norm of the time. Six bits allowed for the upper and lower case letters (56) and the ten numbers, with two left over for special characters – typically a period and a comma. It wasn't until the PDP-11 was introduced that DEC went to eight-bit bytes.

A 12-bit instruction comprised a three-bit operation code, an indirect addressing bit (I), a page-zero bit (Z), and a seven-bit address:



The seven-bit address meant that the PDP-8 memory was organized into 128-word bytes. If the Z bit was set, the referenced address was in page zero. Thus, page zero was a page that was available to any instruction. If the Z bit was not set, then the address referenced was in the current page.

If the I bit was set, the referenced address was the 12-bit address in the next word. This allowed an instruction to reference any of the 4096 addresses in the PDP-8 memory. Therefore, the memory size

was limited to 4096 words. Special instructions were provided to switch memory banks so that a system could be equipped with additional memory.

SAIBOL-8

In the late 1960s,I started my own company, Sombers Associates, Inc. One of the first contracts I landed was to write a PDP-8 payroll program for a company. Doing so in PDP-8 assembly language would have been a near impossible task. DEC had a PDP-8 business language called DiBOL (Digital's Business-Oriented Language), but there was a charge for using it. I wanted to have free use of a business language.

Therefore, I took the time to write a high-level language of my own for the system. It featured a COBOL-like language that I called SAIBOL-8 – Sombers Associates, Inc. Business-Oriented Language for the PDP-8. I was now in a position to write business applications for companies.

The SAIBOL-8 code ran in a second memory bank on the PDP-8. The application program written in SAIBOL-8 ran in the first memory bank and accessed the SAIBOL-8 code in the second memory bank. Having done this, I realized that I could make a business out of running payrolls much cheaper than other payroll companies.

MiniData

I therefore started MiniData Payroll Services, Inc. I leased some space and purchased two PDP-8s. I only needed one, but if I was going to be delivering payrolls, I had better be prepared if one of the computers failed. However, in the more than ten years that I used the PDP-8s, I never had a failure. MiniData ultimately upgraded to PDP-11s.

MiniData advertised through accountants and on billboards. Our billboard message was:

"You pay 6. We pay 9."

The message meant that we would do a nine-person payroll for $6.

MiniData took off, and we were soon doing thousands of small payrolls. The PDP-8 handled the processing load with lots of room to spare.

One of the problems we faced was that the checks had to be personalized for each client with their company name, address, and MICR (magnetic-ink character recognition) encoding. If we had to break down the large printer we used for checks and payroll registers for each company, we never would have been able to accomplish all of the processing.

Therefore, we batched customers into a hundred or so at a time and ran all their payrolls at once. We then separated the payroll registers and checks and added our own MICR encoding to the checks for each customer. We were told that this was illegal, but we never had any problem with it.

Summary

I was involved with the PDP-8 from its very inception and became a close friend of Ken Olsen, the founder of Digital Equipment. The PDP-8 was a computer I loved. With the introduction of SAIBOL-8, I was able to use the system for business data processing and greatly expanded the reach of the system.

The PDP-8 was simple enough that I am able to remember most aspects of it. Computers will never be the same.