Home   About

Search: type, hit enter
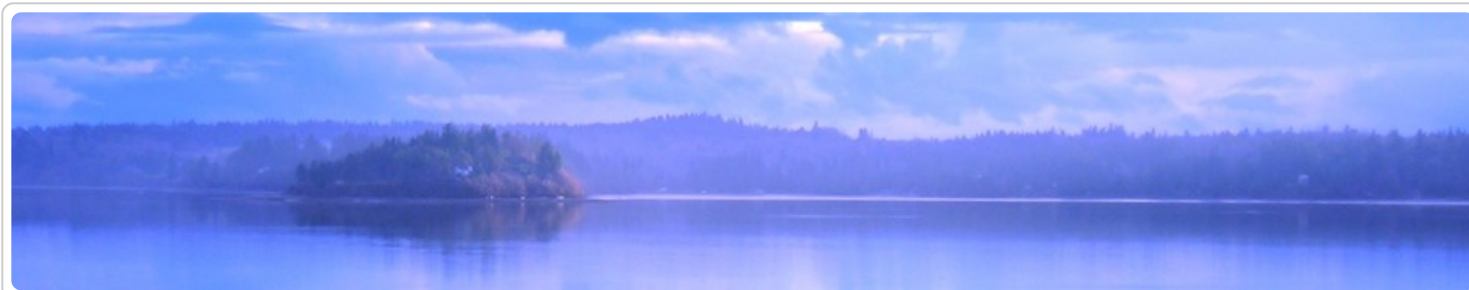
C. Towne Springer   Anything almost right is wrong!

Forth and the AM9511

Posted by admin on February 8, 2015   Go to comments   Leave a comment (0)

While working at the University of Puget Sound in 1982, I was searching for a computing solution for one of the Chemistry researchers who was planning a new instrument for physical chemistry. We looked at PDP-11/xx variants and Data General's smallest systems as well as the latest from HP, which was around $15,000 for a base system. They were all very expensive and carried additional costs for compilers and support. I had a suspicion that an Apple II could do the job but they were also too expensive when you added the risk of not knowing if I could make it work.

I had bought myself a Rockwell AIM-65 6502 computer board in about 1980 to learn about this microprocessor stuff and explore these new possibilities. I was able to teach myself 6502 assembly programming. It also had BASIC, which was quite slow. And I tried the PLM. PLM had a slow development cycle and it was difficult to expand. There were ROMs for a language called Forth, and I got a set. There was a great Forth manual for the AIM-65 and a fine fellow named Gordy Smith who would help with phone support. My AIM had 1K of RAM which was expanded to 4K after by adding a *Little Buffered Mother*, a PCB with slots for hardware expansion. I also bought a board for programming PROMs which allowed expansion of the Forth system in ROM. The AIM-65 has a 20 character LED 7 segment display and a 20 character wide thermal printer. Before long this got to be a handicap and I decided to make the big purchase of a Heathkit/Zenith H8 terminal, and there was an editor for Forth that was easy to use over a terminal. After a lot of messing about with analog and digital circuits to make a better vector display that the World didn't need, I turned my attention to getting more performance from the 6502 and Forth. I made a wire-wrapped expansion board for the AMD AM9511 math coprocessor and wrote a complete interface and math library for Forth. Wow! Some things were more than 100 times faster! And I had fast floating point calculations, integer, single, and double precision, and the conversions. Transcendental functions were also on-chip and a little fancy work with the on-chip stack and I had complex arithmetic and hyperbolic functions.

Ohio Scientific came out with a 6502 based system with a very inexpensive and massive back-plane system that allowed vast expansion. And there was a Forth for it! We got one of these for the new instrument and I began the wire-wrap boars for 12 bit ADC and DAC and the AM9511. This system had 28K of RAM and a dual 8" floppy in a huge separate box. In the end it ran stepper motors, a 4096×4096 storage display, a plotter, shutters, and read data from a photon counter. It did all the data collection and analysis (iterative non-linear least squares fits to multiple exponential decays and phosphorescence peak separation) and plotting and all written in Forth, including the fonts for plotter and storage display — in 24K of RAM!

[About this time one of my physics professors and mentor, F.W. Slee, designed a 6502 based single board computer to use in teaching his electronics class, and we shared a lot of programming and hardware ideas. He independently developed a threaded interpreter much like Forth before ever hearing of it.]

I was able to borrow an Apple II and check the interface. I laid out a circuit board to use the AM9511 in an Apple II/IIe slot and added support for the AM9511 in FigForth and MVPForth on the Apple. I began selling the systems through an ad in Physics Today in March of 1983. I think with ad lead times, I submitted in November or December of 1982 in order to make the March issue.

And here it is. A few years later at Information Appliance, Mino Taoyama pointed out to me that I did not need the slot address switches and decode logic I included. Doh! I had not understood the cleverness of Steve Wozniac's expansion slot scheme, which is quite sophisticated! I think it would have taken only one decoding chip along with the AM9511. The jumper wire is for selecting 2MHz or 4MHz coprocessor. THAT should have been on the rocker switches! Note the nice swoopy-curvey hand layout. Drat! It should have been 3 chips plus the 9511 and a single slide switch or jumper for speed selection!
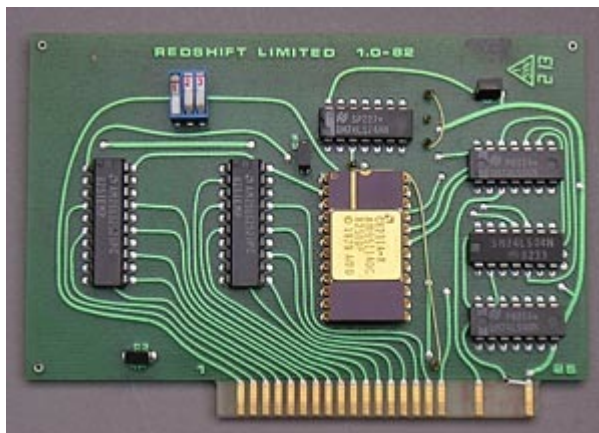
Roy Martens at Mountain View Press was interested in the package and began selling them through his ads in *Forth Dimensions*, the FIG Forth magazine. Forth is an unusual threaded language in which you program by extending the compiler. It is a precursor to modern (non-Lisp) object oriented programming and takes very little space to store programs and is very efficient with RAM usage. Here is the FFPF user manual cover.
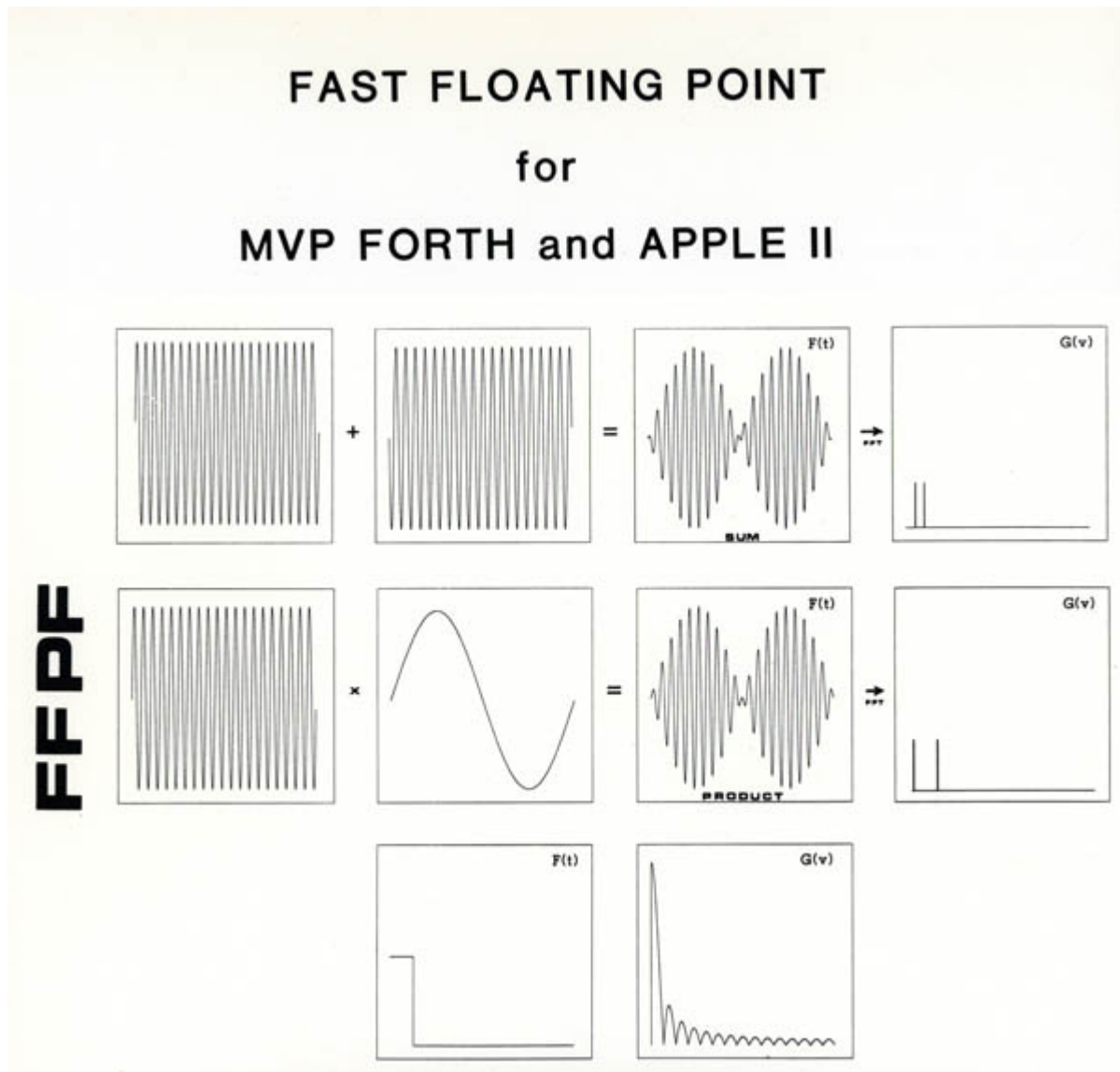
I wrote a plain old FIG Forth with a few Tom Wempe improvements for the ARM processor. The first ARM I had that actually worked was a design by Art Sobel at VLSI. It is a very cool board that had all the peripherals of a PC of the time.
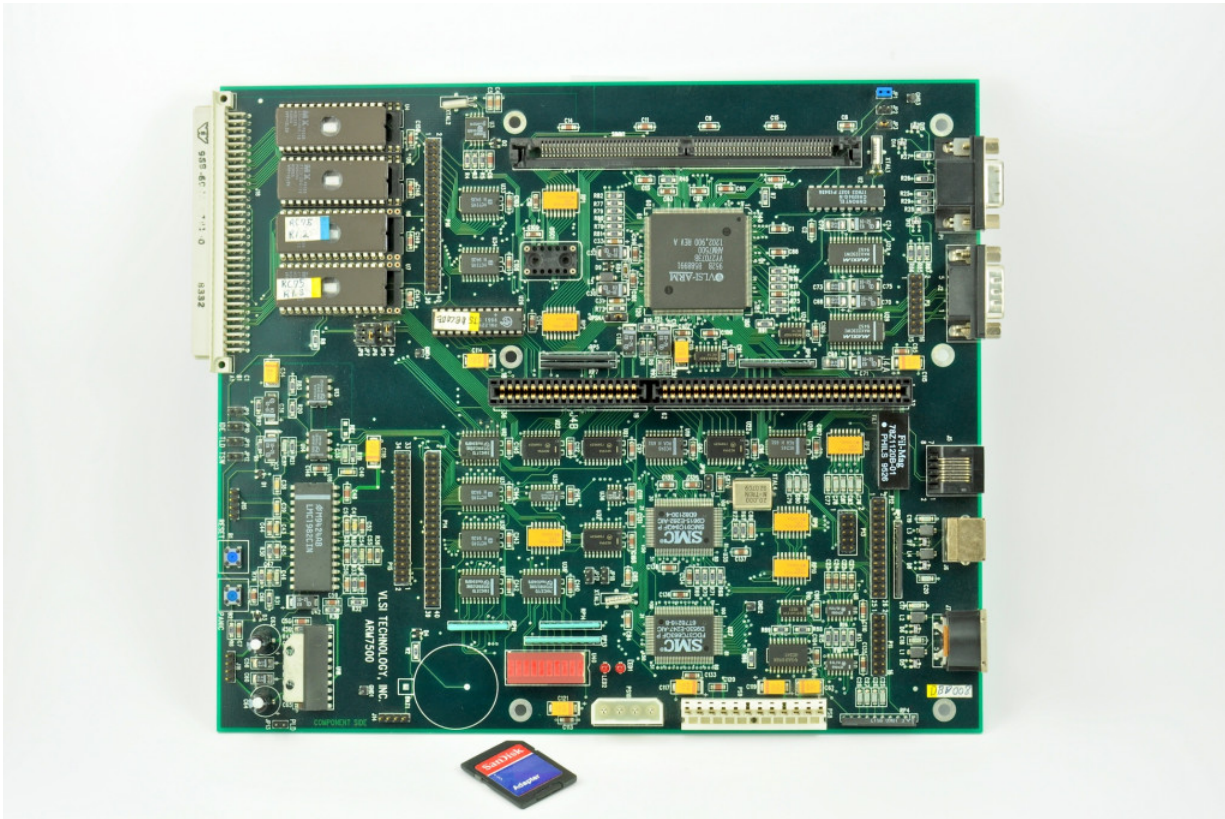
Here it is as I used it a few years later with a CF-II hard drive. 2.5GBytes I think. Lovely board don't you think?

Chapters, FFPF, Forth, uP6502   AIM-65, Apple II, FFPF, Forth, Ohio Scientific

Leave a comment ?   0 Comments.

Leave a Comment

	NAME

| | EMAIL |

| | Website URL |

NOTE - You can use these HTML tags and attributes:
```
<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite="">
<cite> <code> <del datetime=""> <em> <i> <q cite=""> <strike> <strong>
```

S U B M I T

RSS Feed

Search: type, hit enter   SEARCH

Recent Posts

Matrix 3 Axis Accelerometer and FriendlyARM NanoPi.

FriendlyARM NanoPi

MicroPython and STM32F407 Discovery Board

MicroPython and Debian Wheezy

IPython Notebook Examples

Recent Comments

Archives

October 2015

July 2015

April 2015

March 2015

February 2015

Categories

3D Printing

AppleInc

Bio

Chapters

FFPF

Forth

IPython Notebook

MicroPython

NanoPi

uP6502

Meta

Log in

Entries RSS

Comments RSS

WordPress.org