www.UltraTechnology.com

Index | Blog | Streaming Videos | Forth | Chips | Store

UltraTechnology has been out of business since 2002. This site documents its history with code, simulations, documentation, videos, essays, and an ongoing **blog**. UltraTechnology was formed by **Jeff Fox** in Berkeley, California in 1990 to develop parallel Forth computers. In order to best understand the work done at UltraTechnology, and the design work done for UltraTechnology by Charles Moore, the inventor of the Forth programming language, I believe that one needs to understand the context of this work and the meaning of the terms we use when we talk about it.

Context

Charles Moore documented Forth the Early Years at his website. In 1970 he published the first documenation on Forth, Forth - a Language for Interactive Computing where he documents the important concepts of his language. Over the years his language evolved and the terms used to describe it changed, but the original intention of the language was clearly stated as a path to get outside of the multi-language nightmare and into a place defined by one language, Forth, doing everything.

It may be difficult for some people to understand that the goal of Forth was to achieve a improvement in programmer productivity and program performance by having one language. And the fact that Forth was designed to be an OS with cooperative multi-tasker, include a source editor, a compiler, and a command interpreter hides that Forth programs were expected to cooperate with one another. This may be a difficult starting reference point to the generations of programmers raised on the idea that the language (C) was designed to write an OS, UNIX, to protect hostile (C) programs from each other. The fact that things are simpler when one language does everything and programs cooperate may be a difficult starting point for people who brag about their regular use of thirty different programming languges today.

Halfway through the history of Forth its inventor, Chuck Moore declared that the software problem was solved and that the remaining problem was hardware. Forth had been designed to let the programmer change anything and now it was the time to let the programmer eliminate most of the complexity of the compiler and turn the Forth virtual machine into real hardware.

Everything about Forth at this site, and at Chuck's colorforth.com website must be kept in this context. There are people who have used some Forth ideas to go back into conventional environments using Forth in a way that is very different than the context here. This website, and Chuck's website, are not here to say that your version of Forth isn't Forth. Please don't write to me to ask me to have Chuck review your xForth scripting language for the XX Operating System so you can put on your webpage that he approves of it. He doesn't do that. If you want to do what he says is good Forth, or if you want to say something else is good Forth that is fine. This site is not about you, it is about

UltraTechnology and how we did things here.

History

In the sixties I studied AI and later computer graphics and video effects. I got my first PC in 1975. It was the first PC to have a parallel ASCII keyboard input, BIOS in ROM, and video output. I first programmed it using the monitor ROM, later with an assembler, then a BASIC, and in 1978 a FIG-FORTH. That year I also studied digital sound synthesis, advanced machine architecture, and AI at MIT. I did scripting of 3D visualizations and simulations, and wrote varios AI programs for speech, speaker, image recognition, expert systems and learning nets in Forth and later ported from the 8080 to an Apple II. I firmly believed that what would be needed in the future was parallel hardware and what I wanted was something that would run the cooperative style Forth software I had easily and efficiently.

In 1986 I bought a Novix Forthchip Kit from Chuck Moore. It was the simplest computer I had ever used because it was all Forth. The OS, optimizing compiler, and command interpreter was about 6K of object code and 30K of source code. It was smaller and easier to understand than systems with separate BIOS, assembler, compiler, and OS manuals and mutliple languages to learn. I will put the Forthkit manual and Chuck Moore's application notes online in pdf form. I wrote up plans for a parallel computer using multiple Forthchip processors and running a multi-processing Forth OS that would include some learning nets and expert systems features. I made prototypes with old Gavilan laptop parts and Novix and Harris chips and will put some of these 1980s documents online.

I studied expert systems starting with interpreted inference engines useing expert rules in English. The rule interpreter ran a hundred times faster in cmForth on the Novix than it did under an interpreted threaded Forth on my PC. I studied Brad Rodriquiez and other's techniques for compiling expert rules directly into Forth code. But I wanted to get another ten to one hundred times improvement in performance by running code on dozens or hundreds of nodes in parallel.

At the Silicon Valley Chapter of the Forth Interest Group annual Forth Day in 1990 I heard Mr. Moore describe his plan to move to custom VLSI design. He had not been entirely satisfied with the Novix and had made a simpler and 32-bit wide design he had called Sh-Boom that had achieved 80 Forth MIPS in a PGA in 1988. He said the design tools he had used had not understood the unique problems of processor design and that he wanted to write his own tools using Forth. He said what interested him, besides solving the problems of VLSI design and VLSI design tools, was that he wanted to get the order of magnitude increase in speed and two orders of magnitude reduction in cost available by going from programmable logic to full custom VLSI hardware.

I had gone from interpretered expert systems in interpreted Forth on a 6502 where I was getting twenty thousand Forth instructions per second and a few hundred logical inferences per second to native code generated from the same expert rules directly on Novix Forth hardware where one could get millions of logical inferences per second. But I wanted to make each processor a hundred times faster and a hundred times cheaper and then run thousands of those in parallel by going to full custom VLSI parallel Forth chip designs done with Charles H. Moore.

I the early nineties I simulated various combinations of hardware and software as Offete Enterprises' MuP21 and UltraTechnology's F21 (pdf spec) (html docs) design evolved over the years. The original idea was to produce a cheap and simple processor, capable of being a small personal computer, embedded appliance, or node in a scalable multi-processor. The goal was to keep production cost below $1 and make it it fast, low power, and simple to program. F21 was ahead of its time as a chip with a large volume manufacture cost in the nineties of $0.80 and so much on one cheap chip. It included a 220 Forth MIP processor, an independent 40MSPS D/A and A/D coprocessor subsystem, an independent video I/O coprocessor, a smart network routing processor allowing the Forth CPU to execute remote proceedure calls over the network with a one microsecond latency, a real-time clock, parallel port, timer circuit, and 8-bit and 20-bit SRAM and DRAM interfaces in a 68-pin package.

I published simulation results at the Forml Conference in 1991. While surfing the Internet I found Dr. Michael Montvelishsky's paper about teaching parallel Forth in the University in Russia. I wrote to him about ideas in Forth-Linda. Michael published (Parallel Forth - the new approach) in Forth Dimensions about Occam style Parallel Channels in Forth and the simple transition in Forth from cooperative multitasking to multiprocessing using the same wordset. In 1993 I published a paper on F21 and F*F at Forml. And later did an ANS Forth Chess program for F21. I came to consider the concepts in parallel Forth to be about as complex as an "IF" statement. We have continued to work on these and new parallel programming ideas over the years.

In 1995 I joined a NASA startup called the iTV Corporation with Chuck Moore. The idea was to build web appliances with a chip that was a variation of F21, iTV's i21. I became Director of Software at iTV. I trained the staff and we wrote the **4os operating system** and made various all Forth chip and all Forth software web browser and email appliances. After developing products that everyone liked iTV's Board of Directors decided that they didn't want to sell anything and the company was shut down.

By 1999 the money was gone and the financial backers at UltraTechnology had died, gone bankrupt, or pulled out. There was no money for advertizing, promotion, or production of f21. So like iTV's i21 it never went further. The last few prototype chips were sold below cost or given away and things moved on. I got some consulting work on high level AI integration or with F21 prototypes and wrote the aha compiler, OS and GUI. By 2002 UltraTechnology was history except for some sales of a few educational CD and DVD from the online store at times when it was open. I had hoped to move on to making cluster chips, multi-core versions of the parallel Forth chips but my pockets were not that deep.

My idea of a cheap PC had some things in common with MIT's new $200 One Laptop per Child project, except my idea was, "What the world needed was a good $5 computer that was simple to understand." We got pretty close at iTV with the Forthchip module that we could build for about $5 that fit into the battery compartment of a $19 (retail) B/W TV AM/FM which let it connect to the Internet for email and web browsing. iTV never sold any of them let alone the volume needed to sell those miniature Imac-like machines for $49. One of my favorite projects was UltraTechnology's Workstation in a Mouse which showed that a mouse with an F21 would cost about the same as an ordinary $10 mouse but didn't need a PC, it was a PC. It only needed an RGB monitor or TV to play games, develop programs, design chips, do email, or surf the web. iTV did come close to closing a couple of those billion unit type Forth chip Internet Appliance deals.

From 2003 to 2008 I worked with Chuck Moore and people Chuck has worked with for years including John Rible, Dean Sanderson, Greg Bailey, Dr. Michael Montvelishsky, Skip Inskeep, Mark Smeder, Randy Liebernight, Doug Dylan, Charley Shattuck, and other hardware and software engineers using conventional tools or ANS Forth or colorforth software. As System Architect at IntellaSys I worked on hardware, ROM code, development software and applciations. I also trained many new employees on the parallel Forth hardware, software, or Forth CAD tools. It is interesting to see that so many people today are dealing with the issues of going from legacy sequential software to parallel hardware/software that I have been addressing for decades.

More recently I have been working with many of those same people again at Green Array Chips Inc. working in colorforth on new parallel Forth chips.

Some UT video are still available from Google Video in various formats and from this site in the older .rm and smaller .wmv formats for those without access to google video. Skeptics originally notivated me to post actual video of presentations but SVFIG has taken over doing video documentation.

The buttons at the top of the page provide access to the old organization of information at this site, the site might be reorganizated at any time. Please be patient with broken links etc. For the latest information on Chuck Moore's latest Forth hardware/software see **Chuck's colorforth.com blog** and **Green Array Chips**

Listed in the Netscape Directory as a **Cool Site** on the **Forth** programming language.

UT, over a terrabyte served to the public in over 10M hits years ago and still gets about five thousand unique visitors each month.
Site powered by www.strangeGizmo.com