Tangible Work Product

by Thomas Walker Lynch

2022-03-24

Sometimes after reviewing my IP or publications list someone will ask, “but do you do anything?” So here is a list of many things I have ‘delivered’ independent of any considerations of novelty or whether they were patented.

VLSI Chip Custom Design

1. The 29050 SRT divider.
2. The much celebrated 29050 floating-point adder.
3. The 29050 barrel shifter (about half the size of the prior circuit).
4. The multirate clocks on Sonic.
5. The smart card interface on Sonic.
6. VLSI custom logic design of single step count computation and shift, AMD K7.
7. Contributed a lesson for exploring bus controllers to the UT microprocessor lab course.

Software

1. Proprietary JavaScript library for argumentation engine front end, Reasoning Technology.
2. JavaScript portion of the current Reasoning Technology website.
3. Various studies of back end code for Reasoning Technology, using Racket, Ruby, Rust, Common Lisp and C.
4. Lisp program that implements the Birkbeck deep web search algorithms.
5. Lisp code that implements memory interface described in “Tom’s Turing Complete Computer Architecture” book.
6. C++ cycle accurate simulator of RF node processor, Quicksilver Technology.
7. Assembler for RF node processor (lex and yacc), Quicksilver Technology.
8. C++ test bench RF node processor, Quicksilver Technology.
9. System C debugger tool for NetEffect.
10. C++ IEEE 754 elementary functions reference code, Chromatic.
11. C++ IEEE 754 floating-point unit reference code, K6.
12. C++ co-wrote demonstration compiler for High Radix Online Arithmetic.
13. 29000 assembly code, many test vectors for Am29000.
14. Control system language compiler for AMD Fab14.
15. C++ TMYCIN expert system.
16. C++ wrote a Spice Simulator.
17. C++ wrote a program that solved control systems problems.
18. PLS code for part of the cycle simulation of Pine Mainframe for IBM.
19. Bayesian chess player (incomplete, mentioned because it connects to other work).

20. Computer graphics editor - developed over a semester as part of a computer graphics course.
21. Fortran programs for homologous sequence matching of protein structures, UI.
22. Fortran program to draw survey data on Tektronix vector stroke screens, Army Corp of Engineers.

Equipment Builds

1. Built hardware, an actual Neural Network of a dozen neurons using diode dendritic trees, resistor weights, and op amp somas. Wire wrapped on wafer board.
2. Laser light scattering profiler - stepper motor on frame with arm, microcontroller, wire wrapped affair used in another bio lab. Sample went on the stage, arm walked a photo diode, output an amplitude graph.
3. PC peripheral card to catch the boot sequence of a PC and fire off embedded user management software. Prototype was edged PC board holding bus interface and a ROM. It was used by an industrial controls company to prevent machine controllers from being rebooted by those not authorized.
4. Disk head coil on wafer electrical conductivity tester prototype. I designed and built it. Sent the design out for quote for production.
5. UV Blanket Expose standards document for IBM fab.
6. 68000 system board layout. First pass layout for a PCB of new computer.
7. SCSI disk controller. Z80 based design done by another engineer, then I built it. Wire wrap on bread board.
8. Multiple channel pico-probe instrumentation amplifier used in heart research. It was designed by another engineer, and I built it. When finished it fit in a rack mount.
9. Repaired novel presentation equipment rejects at the Mast Keystone factory. These were electromechanical with SSI control so as to sync a controller and a projector.
10. Repaired audio and stage equipment for Boom Audio.
11. Repaired audio equipment for "McNabs"
12. Custom built noise isolation headphones for a bank - first commercial product.
13. Faster method for assembling slide changer mechanism, Mast Keystone
14. Yes, I built some of my ham radio equipment.

Management

1. Business Plan Reasoning Technology.
2. Market Study Reasoning Technology.
3. Business Plan 21st Century Telephone.
4. Top level block design, Gantt charts, test plan, nano-engine AMD K5.
5. Top level block design, Gantt charts, test plan, floating-point unit for K7 - the team reached about 15 when the collapse and purchase of DEC assets lead to a re-org.

Microcode

Here by 'pre-microcode' I mean that the code was written before the hardware simulator was running. For these cases a junior engineer came and coded and tested when the simulator was available.

1. Pipelined BCD conversion on the AM486.
2. Pre-microcode for the K5 floating-point divide program (completed before the hardware simulator was up, a junior engineer actual put the code into the simulator).
3. Co-developed pre-microcode K5 floating-point square root.
4. Pre-microcode for the reduction algorithms and number of terms for the Chebyshev approximations.

Mathematics

1. Formal proofs for all the pipeline predetect cases for the 29050. These were done by hand.
2. Modeled the CS21s process FETs in Mathematica and found best fits for linearized synthesis models.
3. Created a design flow for developing K5 nano-code. This included developing a library for algorithm development and analysis.
4. Algorithm development that came before the microcode projects.

Architecture and Micro-architecture

1. RF node processor, Quicksilver Technology.
2. The K5 nano-code engine, AMD.
3. 29050 team, AMD.
4. TTCA, my architecture proposal.

Studies

1. Bitcoin miner in C. This is part of a study of the bitcoin algorithms and block chain techniques before contemplating custom hardware. My hope was to see optimizations, after all, making fast computation hardware was my occupation at one time. I haven't had time to get back to this in a long time, while bitcoin miners are just getting better and better.
2. Jupyter Lab and Python trader for IBKR - personal study as I am trying to learn more about quantitative finance and algorithmic trading.
3. Wavefront is a compiler for a stackless language. I published the concepts behind it, but did not get very far in writing this with LLVM or in C. This language is for a custom architecture (TCCA) but it should be possible to get it to run on conventional hardware.
4. Endian, a bunch of little things here as proof of concept. It is a simple library for dealing with word endianess, but more importantly it is a working example of headerless C development that came out of the Wavefont work (removes the need for tranche). Also includes a simple tunnel example as an excuse to use the endian swap code.

5. C++, only-one, archive utility that keeps only one copy of duplicate files and an index of their pathnames. Technically this is finished, because it works as intended. However, it raised some issues, and I would prefer it be a filesystem than a utility. The filesystem should be keyword indexed rather than having a directory tree. That automatically takes care of the file duplication without having to create the illusion of multiple files for duplicates. Some good experience with blobs and databases also came out of this.
6. subu - using the linux user group system to act like containers for better security on linux. Mix of Python and C, with C doing the privileged part. Users may create 'sub-users' which they administer. Two sudo bugs were found as part of this. One was the possibility of sending executables through prompts, the other was that sudo zeros out the inherited uid, so the running script can not easily know who it is providing services for. The powers to be elected not to fix this latter one. Also came out of this project was that gdm was forcing the umask on launched processes with world read privilege with no option for changing this. It ignored login.def and everything else. (It now has a pam module that will force it to read the login.def.) I am currently using the method manually, and hope to come back to finish the code. Looks like I need to branch sudo so it won't zero out the effective uid.
7. resources_repo and system are part of a project setup making use of a generic make file. This has been running for quite some time. It continues to evolve as I move between projects in different languages.

Hobbies

1. I organized a small group to go through the Linux From Scratch book. We are just now finishing up.
2. Ham operator KC5TNO.
3. Owner and captain of the offshore sailing sloop "Mystique". We lost Mystique on sand shoaling in Mansfield Pass in 2003.
4. I was in the C league full contact ice hockey in Austin for a season. C league, that is like the league people can play on crutches and still score.
5. Crewed on the museum ship Elissa. That was quite a drive coming up from Austin. Did manage to get on the JC Penny sail.
6. My girlfriend in college gave me the nickname 'never a dull moment' after we built the RC glider 'The Canyon Wren’ then along with some friends took it to the Grand Canyon to settle a bet.
7. Tried to get into race-karts, buddy and I cannibalized a motorcycle and built a kart from scratch, but it did not get past the safety inspector.
8. In high school I did science fairs. Some were fun.
9. I guess putting together the start ups should also be considered a hobby.