Trinary/Midterm Status Report

From Jeff.tk

CAL POLY UNIVERSITY, SAN LUIS OBISPO

MEMORANDUM

DATE: 2008-02-20

TO: Dr. Sema Alptekin

FROM: Jeff Connelly (Computer Engineering), Antonio Chavez (Computer Science), Chirag Patel (Electrical Engineering), advised by Phillip Nico (Computer Science faculty)

SUBJECT: Midterm Status Report on Ternary Computing Testbed

Contents

- 1 Introduction
- 2 Relevant Literature
- 3 Methods, Results, and Discussion
- 4 Future Directions
- 5 Team
- 6 Works Cited

Introduction

As Moore's law reaches its limit, researchers are finding new and innovative ways to increase processor performance. Some techniques, such as multi-core processors, are relatively straight-forward and build upon well-established binary circuit systems, where other technologies such as quantum computing require a radical rethinking of the foundations of computer science. We propose to research a middle-ground by building a functional computer based on trinary logic (base 3). Trinary computers would be digital, just as current binary computers, not as extreme of a change as quantum computers, but new enough to be, for the most part, unexplored territory.

The ultimate end goal of this project is the realization of a physical trinary computer running a simple game. Our hope is to inspire a change in the foundation of computer systems industry which will lead to more efficient systems.

Relevant Literature

We amassed a collection of about 25 documents relating to ternary in some way, many acquired through IEEE. Academic literature on ternary differed in which functions were used, how the ternary digits ("trits") were represented, and how the logic functions were implemented in hardware.

Our first resource, Steve Grubb's web site[1], defined unary (1-input) functions denoted by symbols ↘, ↗, ∩, ∪, and /, respectively, denoting shift down, shift up, rotate up, rotate down, and invert. Grubb used unbalanced ternary with digits 0, 1, and 2 in his logic circuits. Transistor models [2] implemented these unary logic gates using from 5-10 resistors and 2-6 Bipolar Junction Transistors. Grubb also implemented the dyadic (2-input) functions MIN and MAX, analogous to boolean AND and OR. The logic functions that Grubb gives produce a complete system.

Mouftah[3] also used unbalanced ternary, but defined vastly different unary gates, denoted by $\overline{x^0}$, $\overline{x^1}$, $\overline{x^2}$, $x^{\neg}$, and $x^{\lceil}$ for negative/simple/positive ternary inverters, forward diode, and reverse diode. Combined with the dyadic MIN and MAX functions, these unary functions build a complete system. Mouftah's inverters consist of two MOSFET transistors and two resistors, and the forward/reverse diode gates consist of nothing more than a diode and a shunt resistor. Mouftah's ternary algebra is simpler, as the three inverters all obey a base-three analogue of DeMorgan's law. We developed an alternate symbols for representing Mouftah's gates in balanced ternary, and his gates are currently the most likely for us to construct. However, Mouftah did not include a "symmetric" unary gate, so not all unary functions can be created only with his unary gates--for some functions, a dyadic gate is required. We are investigating possible symmetric unary gates to fill this gap.

Papers on trinary minimization [4] [5] [6] [7] often uses a third set of unary functions, denoted as x^0, x^1, and x^2 in unbalanced ternary. Transformations can be made between these functions and Mouftah's functions. Our wiki page[8], incorporates ideas from each of these three systems of ternary logic. All the systems have in common their definition of the MIN and MAX dyadic functions. A fourth article we researched, the TriINTERCAL manual, defined "ternary OR" in the same way as MIN, but "ternary AND" differently, in addition to defining a BUT dyadic function. The manual also defined an inconsistent mapping between balanced and unbalanced ternary. We did not adapt these conventions.

At a higher layer of abstraction above the gate level, Knuth[9] wrote two pages on balanced ternary, including the history and mathematically-convenient properties of trinary. According to Hayes[10], if hardware cost is measured as the product of the radix times the numerical width, then "ternary is inferior to binary only for 8,487 values of rw; ternary is superior for infinitely many values." because it is closer to Euler's number, *e*.

In our research, we found numerous references to Setun, a Soviet computer built in the late 1950s[11]. Fifty Setuns were constructed before binary became the dominant paradigm. A software-emulated machine, dubbed Ternac, was developed at State University of New York by Gideon Frieder in 1973. Frieder published an article on the benefits of ternary computers[12] and an efficient method of emulating ternary on a binary computer[13].

Additional literature was gathered and stored at [14].

Methods, Results, and Discussion

All of our work is available published on a customized MediaWiki web site[15]. Detailed weekly updates on our progress, recorded individually, is available[16].

Jeff heavily revised and expanded an older trinary research paper and published it on the wiki. It is composed of several sections:

- Base 3[17], discussing aspects of the numerical system itself (7 pages)
- Logic[18]., all about unary and dyadic trinary functions (17 pages)
- Arithmetic[19], a shorter page on balanced trinary arithmetic (5 pages)
- Implementation[20], concerning the hardware design of trinary logic (10 pages)
- Architecture[21], design of the trinary computer system (8 pages)

These sections, totaling 47 pages, are the primary result of this project at this time and are publicly available on the Internet[15].

Antonio and Chirag are working on simulating the hardware and software aspects, respectively. Currently, Antonio's simulator is 545 lines of Python code and the tokenizer and parser are functional. According to David a Wheeler's SLOCCount program[22], the 371 physical lines of code (excluding blank lines and comments) are estimated to have taken 0.85 person-months of effort, and 2.35 months to complete. This is is line with the amount of time currently spent on the project.

Antonio has learned a new computer programming language (Python) and has familiarized himself with VHDL with the help of Jeff to develop the simulator. VHDL was used as a model for the simulator language. The tokenizer is complete and a unary expression parser is also included. The source code is stored in a git repository for version control, publicly mirrored and available at: http://repo.or.cz/w/trinary.git

Chirag has simulated Grubb's BJT-based inverter and rotate down gates discussed earlier. Full results are not yet published. Chirag also completed schematics, simulations and analysis of Mouftah's basic ternary gates[3] including TOR, TAND, TNAND, TNOR, STI, PTI, NTI, EPTI, ENTI, EPTNAND, ENTNAND, EPTNOR, ENTNOR. Chirag is currently further analyzing these ternary gates and considering alternatives. Physical hardware testing of these gates will be conducted once funding is available to purchase parts. The hardware portion of the project is currently in the research phase (Schematics and Simulations). All the simulations and schematics will soon be available on the Wiki along with detailed analysis.

Future Directions

We have a general timeline available[23] and also a specific to-do list for each team member[24]. That said, our future plans are to wrap up the literature research and begin to construct, simulate, and test trinary circuits after funds are obtained to purchase the necessary equipment.

First we will develop and test basic trinary logic gates. Next, we'll focus on building multiplexers, and then a simple ternary switch-based ROM. When this it accomplished, we will move onto the other components of the computer architecture[25].

Our schedule is threatened by the lack of funding we have not yet been able to receive, as of the sixth week of the quarter. With only 20 weeks from start to finish for all the funds to be expended, 30% of the available time has passed without funding. We expect to receive funding soon in order to purchase parts required for circuit construction and analysis.

Team

Jeff Connelly: Researching the architecture of a trinary computer system to bridge the hardware with the software.

Antonio Chavz: Researching the changes that will be made to software to support a trinary computer system and developing a software simulator.

Chirag Patel: Researching the electronic circuitry that will be necessary to implement to realize the trinary computer system.

Works Cited

Note, an online copy of this report is available at http://jeff.tk/wiki/Trinary/Midterm_Status_Report for your convenience.

1. ↑ Grubb, Steve. Trinary.cc, Available: http://trinary.cc/
2. ↑ Grubb, Steve. Trinary.cc - Transistor Models, Available: http://www.trinary.cc/Tutorial/Transistors/Transistors.htm
3. ↑ [3.0] [3.1] study on the implementation of three-valued logic H. T. Mouftah May 1976 Proceedings of the sixth international symposium on Multiple-valued logic, Available: Image:P123-mouftah Study on the Implementation of Three-valued Logic.pdf
4. ↑ D.I. Porat, "Three-valued digital systems", Proc, lEE, Vol. 116, No. 6, June 1969, pp. 947-954.
5. ↑ M. Yoeli and G. Rosenfeld, "Logical design of ternary switching circuits", IEEE Trans. Elect. Comp., Vol. EC-14, February 1965, pp. 19-29.
6. ↑ M. Bitran and M.J.O. Strutt, "Minimization of ternary logic and complete set of integrable circuits", Electron. and Commun., AE0, Band 25,No. 8, 1971, pp. 387-392.
7. ↑ R.S. Nutter and R.E. Swartwout, "A ternary logic minimization technique", Conference Record of the 1971 Symposium on the Theory and Applications of Multiple-valued Logic Design, May 1971, pp. i12~125.
8. ↑ Trinary/Logic - Jeff.tk, Available: [[1] (http://jeff.tk/wiki/Trinary/Logic)]
9. ↑ D.E. Knuth, The Art of Computer Programming - Volume 2: Seminumerical Algorithms, pp. 207-208. Addison-Wesley, 3rd ed., 1998. ISBN 0-201-89684-2. Available: Image:Knuth-TaoCPVol2-pg207,8.pdf
10. ↑ American Scientist: Computing Science: Third Base by Brian Hayes, 2001. Available: Image:American Scientist Online - Third Base.pdf
11. ↑ Setun' W. H. Ware, S. N. Alexander, N. M. Astrahan, H. H. Goode, M. Rubinoff, P. Armer, L. Bers, H.d. Huskey, "Soviet computer technology - 1959," Communications of the ACM, pp. 149-150, 1960. Available: Image:Communications of the ACM - Soviet Computer Technology - 1959.pdf
12. ↑ Frieder, G. and Luk, C. Part 1 - Motivation for Ternary Computers, 1972. Available: Image:P83-frieder Ternary Computers - Part 1 - Motivation for Ternary Computers.pdf

13. ↑ Frieder, G. and Luk, C. Part 2 - Emulation of Ternary Computer. 1972. Available: Image:P86-frieder - Ternary Computers - Part 2 - Emulation of Ternary Computers.pdf
14. ↑ Available: Trinary/Literature or http://jeff.tk/wiki/Trinary/Literature
15. ↑ [15.0] [15.1] Trinary - Jeff.tk. Available: http://jeff.tk/wiki/Trinary
16. ↑ Trinary/Status - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/Status
17. ↑ Connelly, Jeff et al. Trinary/Base_3 - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/Base_3
18. ↑ Connelly, Jeff et al. Trinary/Logic - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/Logic
19. ↑ Connelly, Jeff et al. Trinary/Arithmetic - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/Arithmetic
20. ↑ Connelly, Jeff et al. Trinary/Implementation - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/Implementation
21. ↑ Connelly, Jeff et al. Trinary/Architecture - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/Architecture
22. ↑ Wheeler, David A. SLOCCount. Available: http://www.dwheeler.com/sloccount/
23. ↑ Chirag, Patel et al. Trinary/Timeline - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/Timeline
24. ↑ Trinary/TODO - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/TODO
25. ↑ Connelly, Jeff et al. Trinary/Architecture - Jeff.tk. Available: http://jeff.tk/wiki/Trinary/Architecture

Retrieved from "http://jeff.tk/wiki/Trinary/Midterm_Status_Report"

- This page was last modified 01:53, 21 February 2008.
- Content is available under GNU Free Documentation License 1.2.