Trinary/Final Status Report

From Jeff.tk

Contents

- 1 Executive Summary
- 2 Introduction
- 3 Background
- 4 Design, Theory, Methods, and Experimentation
  - 4.1 Hardware
    - 4.1.1 Summary
    - 4.1.2 Simulations
    - 4.1.3 Architecture
  - 4.2 Software
- 5 Conclusions
- 6 Acknowledgements
- 7 Works Cited

Executive Summary

The end goal of our project is to build a simple base-3 (trinary or ternary) computer to execute a simple game. To achieve this goal, we first spent the beginning of the quarter researching and reading published articles on trinary computing, determining the best approach to build logic gates for our computer. We were impressed by the elegance of the logic circuits designed by Hussein Mouftah, and decided to use his gates in our project. We successfully simulated 10 MOSFET-based trinary logic gates in PSpice and published the results on our wiki at http://jeff.tk/wiki/Trinary. Next, we designed an incipient computer architecture, while developing trinary simulation software. This quarter's work sets the stage for construction and simulation of the trinary computer in the quarter to follow.

Introduction

As Moore's law approaches its limit, researchers are finding new and innovative ways to increase processor performance. Some techniques, such as multi-core processors, are relatively straightforward 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 the benefits of a middle ground by building a functional computer based on trinary (base 3) logic. Trinary computers are digital, as are binary computers—they are not as extreme of a departure from existing computers, but are relatively 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 leading to more efficient systems.

Background

The earliest computers were analog, and the first digital computers were base 10, just as the decimal numbering system commonly used today. Shortly, binary computers became the norm as George Boole's algebra was found to be useful in their construction. Other radixes, including -2 and 3, were briefly explored -- a trinary computer named Setun was built in Russia in 1959 but abandoned because it was seen as unconventional. Trinary, however, has several interesting properties. From trinary naturally follows "balanced ternary arithmetic", a numbering system using digits -1, 0, and 1, which Donald Knuth (sometimes known as the Leonard Euler of Computer Science) called the "the prettiest number system of all" [1]. Balanced ternary easily represents negative and positive integers without the need for a sign bit, as required in binary computers today.Trinary can naturally be represented at the circuit level using negative, low, and positive voltages. This makes trinary a more attractive radix than higher values, which would cause difficulty in distinguishing between the valid states. Trinary also has theoretically optimal properties, being the closest integer to e[2], as well as superior round-off properties[3].

After extensively researching existing literature on trinary, we decided on a set of gates designed by Mouftah rather than Grubb or Yoeli. A full literature summary is available in the midterm status report[4] and will not be repeated here.

Design, Theory, Methods, and Experimentation

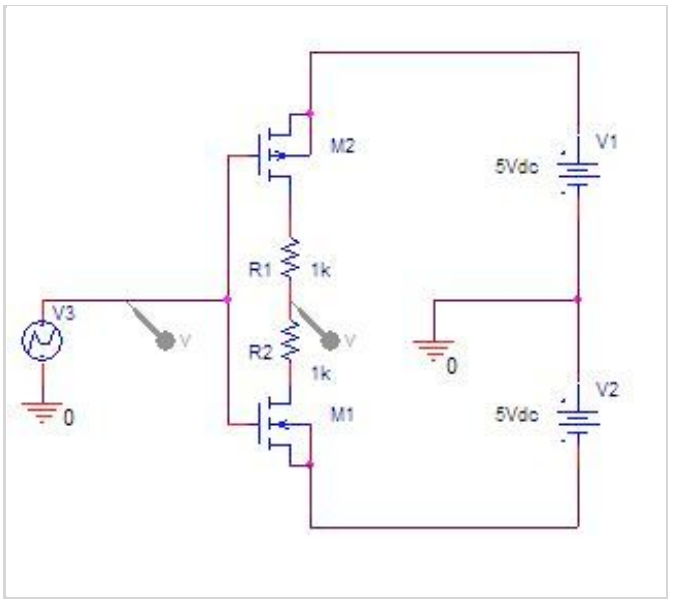
Hardware

Summary

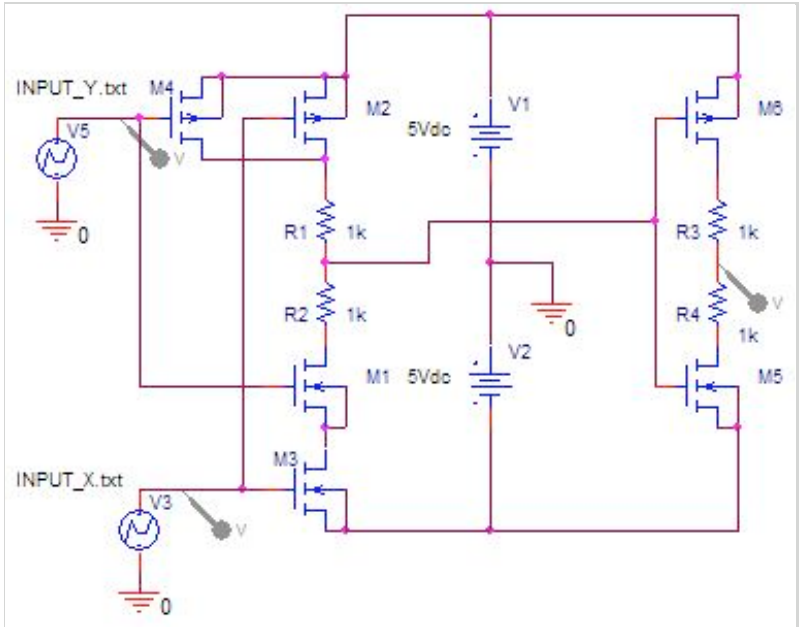
This quarter, we have made significant progress on the design and theory of various circuits for the Trinary Gates. For further design we are currently in the process of obtaining OrCAD Capture from Cadence through an Academic licensing program. Once this software is secured we will be able to continue simulation of circuitry and also part design (alternatively, we may use the fully-functional and freely-available LTSpice (http://www.linear.com/designtools/software/) simulator and capture program). The architecture will be simulated by creating higher level parts allowing for a full analog simulation in parallel with physical circuit construction. We recently received a shipment of parts for our project and we plan to begin building the computer (gates, multiplexer, etc.) at the beginning of Spring 2008. Simulation and construction will continue through Spring 2008 along with a trinary expression evaluator being developed by Antonio Chavez. The gates are designed to use MOSFETs and we have decided to use the CD4007 quad complementary MOSFET pair integrated circuit for our simulation and physical hardware. Also, we have purchased tri- and bi-color LEDS, triple-throw switches, transmission gates, breadboards, jumper wires, battery holders, and batteries. All of the latter parts will be used for construction of multiple copies of trinary gates on separate breadboards, and combinations of these gates will be used to make the computer. Memory and user input will be programmed using the triple-throw switches.

Simulations

We simulated the PTI, EPTI, NTI, ENTI, STI, TAND, TOR, TNAND, and TNOR gates. Selected schematics have been provided below. The circuits below were originally designed by Mouftah[5]. The gates listed above along with Forward Diode and Reverse Diode will provide us the necessary gates for construction of any ternary function. Once these gates have been physically constructed and their operation verified, we will be able to move on with construction of higher level parts from these basic gates. Several new parts (models) have been constructed in Cadence Allegro Design Entry CIS for re-use in higher level simulation, which will later be reconstructed using the new software either obtained from Cadence or from Linear Technology.



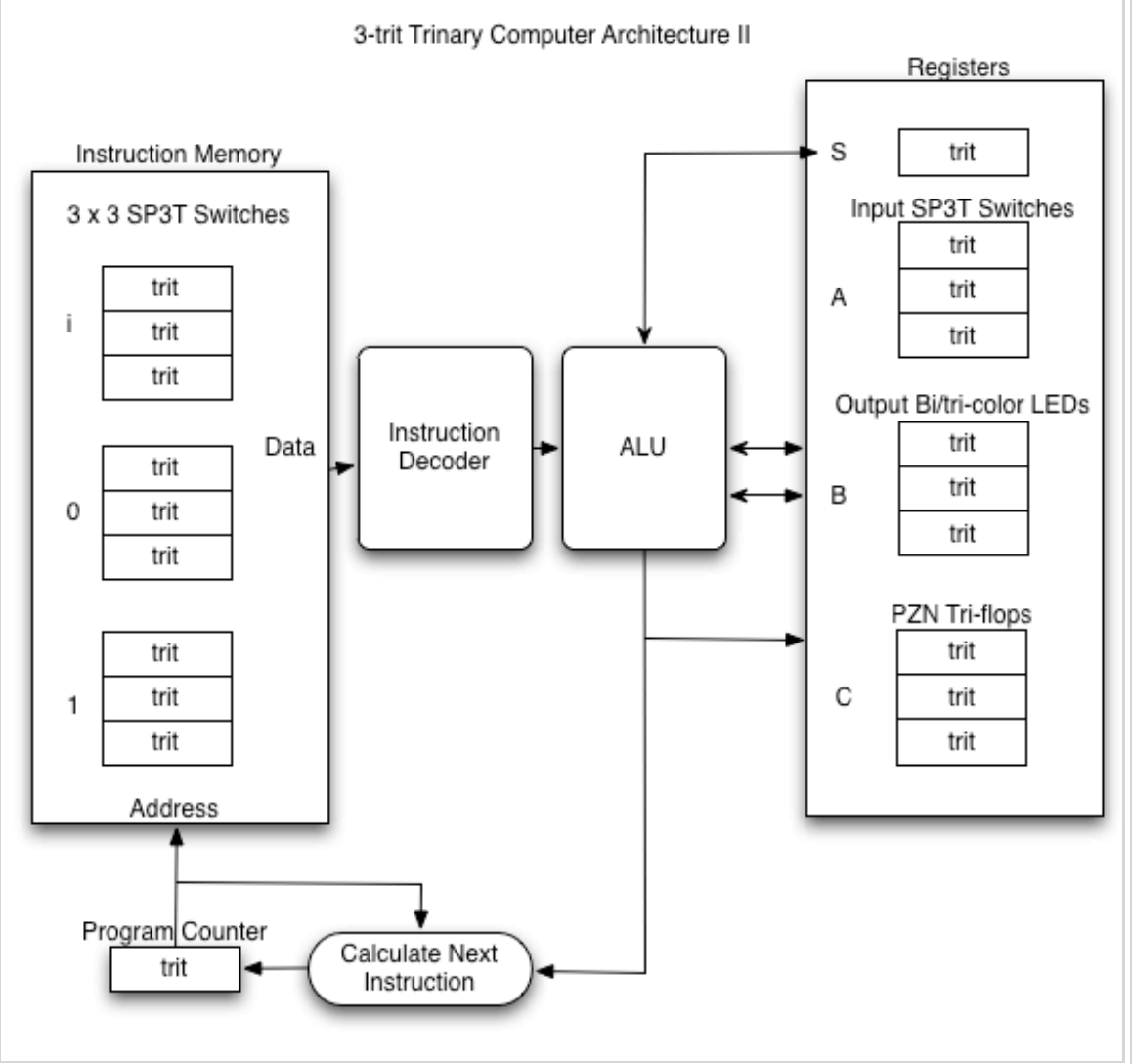
Simple Ternary Inverter Schematic



TAND Schematic

Architecture

Jeff has continued to develop the computer architecture as well as integrate with Chirag on transistor level design and development. Our latest proposed architecture is:



Block diagram for 3-trit trinary computer architecture, version 2.

The proposed Instruction Set Architecture (ISA) includes three opcodes, to compare, store, and conditionally branch. Using each of these three instructions, Jeff wrote a simple 3-instruction guessing game[6]. He then wrote a 100-line assembler in Python, supporting each of these instructions (and labels), assembling the program into the tritstream sequence $0\bar{1}01\bar{1}11\bar{1}0$. This program can be tested once a simulator is written.

Software

Antonio has been researching and developing a trinary logic simulator that is influenced by the VHDL model. This development process required some initial overhead due to the need to learn a new programming language (Python), learn VHDL, and to research the best approach to develop the simulator. Details on the research conducted towards the development can be found at http://jeff.tk/wiki/Trinary/Digital_Simulation. The source code repository for the simulator can be found here: http://repo.or.cz/w/trinary.git.

After working on the VHDL-like simulator for most of the quarter, we decided to instead use analog PSpice simulation for hardware, as it offers superior accuracy and will be more helpful in building the circuit. For this reason, after this quarter Antonio will shift focus to a higher-level simulation, simulating at the instruction-

level as well as writing an arbitrary trinary expression evaluator.

Conclusions

During the course of our research we were able to find additional supplemental mathematical proofs of the advantages of base three numbering systems. The dyatic gates we plan to develop allow for backward compatibility with existing software and in addition the applications will benefit from the increased performance of these trinary-based dyatic gates. Furthermore, the trinary foundation of computing will encourage the development of innovative applications that will take advantage of balanced trinary arithmetic and the introduction of an "unknown" state to the existing "true" and "false" states.

Acknowledgements

Phillip Nico - Advisor Support

Cynthia Bitto - Finance Support

David Braun - Circuitry Support

Sema Alptekin - Honors Program Support

Works Cited

1. ↑ 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.
2. ↑ American Scientist Online, November-December 2001: Third Base, Brian Hayes
3. ↑ Ternary computers: part I: motivation for ternary computers, International Symposium on Microarchitecture archive, Conference record of the 5th annual workshop on Microprogramming table of contents, Urbana, Illinois, 1972
4. ↑ J. Connelly, A. Chavez, C. Patel, Trinary - Midterm Status Report. Online, Available: http://jeff.tk/wiki/Trinary/Midterm_Status_Report
5. ↑ A 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
6. ↑ Connelly, Jeff. guess.t, Available: http://repo.or.cz/w/trinary.git?a=blob;f=asm/guess.t

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

- This page was last modified 09:50, 14 March 2008.
- Content is available under GNU Free Documentation License 1.2.