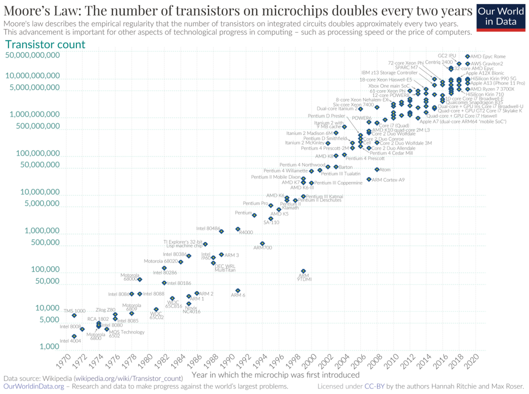
A Comparative Analysis of the Performance of Emulated Ternary vs Binary Logic Systems

Project Proposal

Thomas Timmons
Bachelor of Science in Computer Science
The University of Bath
November 8, 2024
`tlt33@bath.ac.uk`

1 Problem Description

In 1965, Gordon Moore, made an observation that density of transistors within integrated circuits grow exponentially over time and predicted that it would double every two years. This later became known as Moore's Law and almost 60 years later, this still holds true.



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.
Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Figure 1: Moore's Law

The implication of Moore's law is that the performance of digital systems has also increased exponentially over time. (**?**) Many of the devices we use today, reap the benefits of this increase in performance. Smaller and faster computers have allowed for improvement in industries such as health care, education, transportation and along with many others that have helped create a higher standard of living for many people around the world.

However, researchers have begun to question the sustainability of Moore's Law as the size of transistors approach the atomic scale and the physical limits of Moore's law should be reached at some point in the 2020's. (**?**)

Even Moore himself was quoted in 2006 saying "...the fact that materials are made of atoms is the fundamental limitation and it's not that far away...We're pushing up against some fairly fundamental limits so one of these days we're going to have to have to stop making things smaller". (**?**)

This has led to the exploration of alternative computing paradigms such as quantum computing.

1.1 Ternary Algebra

To represent any given range of values N, with the radix of value is given by R and the necessary number of digits required is d (rounded upwards to the nearest integer):

$$N = R^d \tag{1}$$

The cost or complexity of the system C is directly proportional to the capacity of the digits $(R \cdot d)$. Therefore for some constants k and $\log N$: (**?**)

$$C = k(R \cdot d) = k(R \cdot \frac{\log N}{\log R}) = k \log N(\frac{R}{\log R}) \tag{2}$$

Minimising this cost C can be done by differentiating with respect to R and setting the result to 0:

$$\frac{\partial C}{\partial R} = k \log N \cdot \frac{d}{dR}\left[\frac{R}{\log R}\right] = k \log N \cdot \frac{\log R - 1}{(\log R)^2} = 0 \tag{3}$$

After removing constants $\log R = 1$ and solving for $R = e = 2.718$. Since as stated above, the radix R must be an integer then we are left with $R = 3$. Therefore, the ternary logic system is the most efficient. (**?**)

In binary logic the binary set of values is $\mathbb{B} = \{0, 1\}$ while the symbol $\mathbb{T}$ will be used to denote the ternary set. There exists a variety of valid ternary sets with the most natural being an extension of binary called unbalanced ternary where $\mathbb{T} = \{0, 1, 2\}$. Unknown state ternary where $\mathbb{T} = \{T, ?, F\}$.

In this paper we will be using balanced ternary where $\mathbb{T} = \{-, 0, +\}$. This will allow operations on negative and positive numbers without having to use the ternary equivalent of two's complement.

1.2 Aims

Researchers have attempted to investigate this field using two distinct approaches. One from a more theoretical perspective or looking at the theoretical gains of ternary logic while others taking a more practical approach. Ternary test benches have been created in order to attempt to validate the theoretical performance gains.

However, the underlying implementations of the basic ternary logic is underpinned by Complementary Metal Oxide Semiconductor (CMOS) along with PMOS and MNOS.

For this project I propose several theoretical gates whose logic can be implemented in hardware that have the same relative speed to their binary counterparts. More formally we assume the following is true in regards to propagation delay:

$$\begin{aligned} AND &\equiv \mathbb{T}AND \\ OR &\equiv \mathbb{T}OR \\ NOT &\equiv \mathbb{T}NOT \end{aligned} \tag{4}$$

This will allow us to create a basic binary and ternary emulator whose performance will be directly comparable.

The aims for this project are too:

- Collate research around how ternary logic can be implemented and the equivalence of a variety of operations across the two bases.
- Build two emulators with both binary and ternary logic respectively with an equivalent architecture.
- Practically test the theoretical benefits of ternary logic over it's binary counterparts.

1.3 Objectives

There are 3 main objectives that must be reached:

1. Build a binary emulator with a very basic CPU architecture and simple instruction set.
2. Build a ternary emulator with an equivalent CPU architecture and instruction set.
3. Write a simple complier to create code to run on both machines.

2 Requirements Specification

2.1 Metrics

The performance of various CPU's can be measured using the following criteria:

- Power Consumption
- Cost
- Maximal Frequency
- Instructions Per Cycle (IPC)
- Architecture Efficiency
- Number of Cores
- Cache Size and Speed
- Thermal Efficiency

We will only be considering the maximal frequency for these systems as power consumption, cost and thermal efficiency are hard to deterministically measure within an emulator. The other possible metrics are discussed in the optimisations section.

The maximal frequency, also know as clock speed, of a CPU represents the number of cycles it can execute per second, typically measured in Hertz in the order of GHz. However, we will be measuring the time taken for the signal to propagate through the circuit. This is determined by the longest path of the electric signal known as the critical path. Using our theoretical gates stated above (1) we can simply calculate the propagation time for any given circuit by the number of gates required to perform any given operation.

Time taken for delay in wires will not be taken into consideration.

All base gates will have an arbitrary unit for propagation time of 1 whereas more complex gates will have greater propagation times.

2.2 Optimisations

Binary systems have had decades of research and optimisation over their ternary counterparts. The aim of this paper is not too optimise ternary circuits and will therefore not be implementing optimisations in either circuits as we are only concerned with the relative performance.

In reference to the criteria above, only the the architectural efficiency of the emulator will be considered as both will only have a single core with no cache.

2.3 Requirements

Below is a detailed set of requirements needed to meet the objectives set above and in turn fulfill the aims of the project. There will be categorised by the MoSCoW prioritisation method as shown below:

1. **Must** - Non-negotiable requirement.
2. **Should** - Important but not vital to the outcome.
3. **Could** - Desirable but only to be done if their is extra time.
4. **Won't** - Out of scope of the project due to time constraints.

2.3.1 Functional

- Both emulators **must** be built in Rust due to a lack of flexibility within emulating tools due to being so closely tied to binary logic.
- Both emulators **must** have an arithmetic and logic unit, control unit, program counter and and registers within the CPU.
- Both emulators **must** have a comparable amount of memory although it does not need to be exact.
- Both emulators **must** be able to measure the propagation delay for any algorithm performed in the emulator.
- Both emulators **should** be able to handle floating point arithmetic.
- A testing workbench **must** be implemented in order to be able to record interactions with the emulator simultaneously.
- Design and building a compiler **should** be implemented to allow for fast test cycle.
- Cache **won't** be implemented on either emulator as won't affect comparability.
- A system kernel **won't** be implemented on either emulator as it is outside of the scope of the project.

2.3.2 Non-Functional

- A Literature, Technology and Data Survey **must** be completed by the 6th December 2024.
- A basic implementation of both emulators must be implemented for the demonstration of progress on the 17th February 2024.
- The project **must** be completed by the 2nd May 2025.

The above requirements are not fixed and are subject to change as the project progresses and develops.

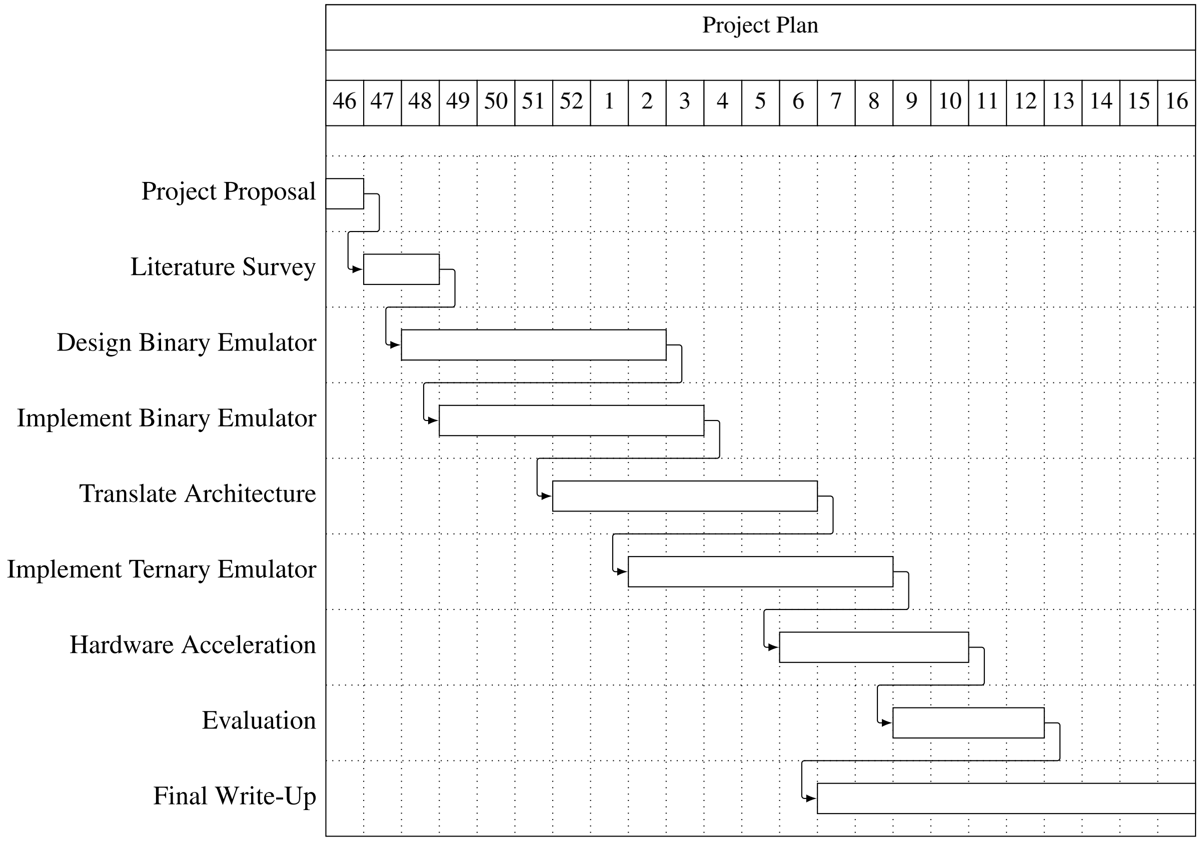
3 Project Plan

3.1 Deliverable Deadlines

Deliverable	Date
Project Proposal	08/11/24
Literature, Technology and Data Survey	06/12/24
Demonstration of Progress	17/02/25
Dissertation	02/05/2025

3.2 Initial Gantt Chart

Task	Start Date	End Date	Effort (Hours)
Project Proposal	10/11/24	17/11/24	20
Literature, Technology and Data Survey	18/11/24	29/11/24	20
Research and Design Basic Emulator	30/11/24	15/01/25	60
Implement Binary Emulator in Rust	7/12/24	30/01/24	75
Translate Binary Architecture into Ternary	01/01/25	15/02/25	100
Implement Ternary Emulator in Rust	15/01/25	30/02/25	40
Implement hardware specific circuits	15/02/25	15/03/25	20
Evaluation	15/03/25	01/04/25	40
Final Write-Up	15/02/25	02/05/25	100



The project plan has been designed around the immoveable deadlines while incorporating a large margin for particular parts of the project overrunning. It also takes into consideration the parallel nature of the tasks where designing and implementing the emulators can be done with a bit of overlap in order to test different architecture and design decisions throughout the project.

4 Resources