Ternary Computer Architectures

Vega Carlson
Computer Engineering
University of Nebraska - Lincoln
Ashland, USA
vegac@protonmail.com

***Abstract*—Ternary logic, that is logic with three states instead of the traditional two found in binary, is currently being explored as one option to help surpass performance barriers faced by traditional architectures as processor clock speeds and node sizes become more difficult to improve. This paper attempts to collect, summarize, and glean new insights from both old and recently published research papers that concern the future of ternary computer architectures and logic systems.**

***Index Terms*—Ternary Logic, Multiple Valued Logic (MVL)**

I. Introduction

Ternary logic is a concept that seems almost foreign in today's digital landscape, where the simplicity of binary is ubiquitous. Bringing it up at all seems to be some sort of challenge to the status quo, but what is it? Ternary logic is simply logic which uses three states instead of the traditional two in a binary system. In binary logic, the only obvious choice for these states is True and False, the one and zero that all computer scientist know. Ternary, on the other hand, can come in multiple forms, as can be seen in Table I, this alone makes research into the topic more tedious if one wants to explore the idea in full. Furthermore, due to the ubiquity of the binary computer and how well understood they are, there is less motivation to study multiple-valued logic (MVL) systems to begin with. Regardless, advancements are still being made in the field. Recently, many of these advancements have been in systems utilizing systems not based on silicon but instead technologies such as carbon nanotube field effect transistors (CNFETs) or optical computing.

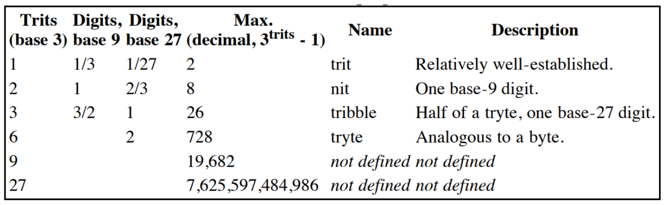
One problem when studying ternary is that there is generally a lack of standardized terminology for things. When necessary this paper will use the terminology in Figure 1 as laid out by Douglas Jones in *The Ternary Manifesto* [2].

The primary motivation to study MVL systems comes from the need to surpass the barriers that are being faced in traditional architectures: frequency scaling and process node limitations. Anybody interested in buying a processor in the last decade can attest to the stagnation of frequency on the highest end components, with around 5Ghz being the maximum most prosumers can expect to achieve without needing to artificially improve their odds in the silicon-lottery by looking for particularly well binned silicon. Similarly, the year by year shrink in processor node sizes has been assumed for a while now to be near its end as the effects of quantum tunneling mean that smaller nodes cease to be possible without spontaneous bit flips.

Before investing the history and future of ternary, it helps to have some background knowledge on at least one ternary logic system. Despite the initial strangeness of it, unknown state logic, that is logic with true, false, and "maybe" states, is actually one of the easier to explain. Put in a rather reductive way, the "maybe" state simply propagates though. For example, an "AND" gate operating on binary outputs true if and only if both inputs are true. While obvious from that description, this also

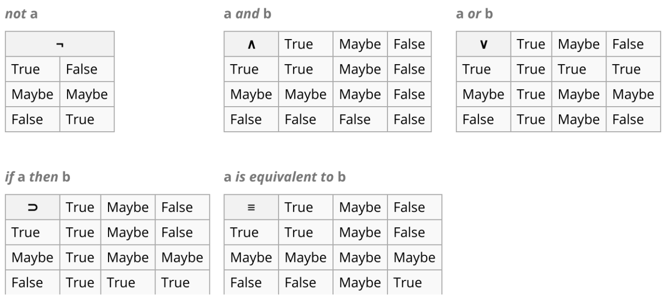
TABLE I
TERNARY LOGIC SYSTEMS [1]

Set	Name / comments
0,1,2	Unbalanced Trinary
0,$\frac{1}{2}$,1	Fractional Unbalanced Trinary
-1,0,1	Balanced Trinary
F,?,T	Unknown State Logic
T,F,T	Trinary Coded Binary



Trits (base 3)	Digits, base 9	Digits, base 27	Max. (decimal, 3^{trits} - 1)	Name	Description
1	1/3	1/27	2	trit	Relatively well-established.
2	1	2/3	8	nit	One base-9 digit.
3	3/2	1	26	tribble	Half of a tryte, one base-27 digit.
6		2	728	tryte	Analogous to a byte.
9			19,682	*not defined*	*not defined*
27			7,625,597,484,986	*not defined*	*not defined*

Fig. 1. Names of ternary numeric resolutions



not a

¬	
True	False
Maybe	Maybe
False	True

a *and* b

∧	True	Maybe	False
True	True	Maybe	False
Maybe	Maybe	Maybe	False
False	False	False	False

a *or* b

∨	True	Maybe	False
True	True	True	True
Maybe	True	Maybe	Maybe
False	True	Maybe	False

if a *then* b

⊃	True	Maybe	False
True	True	Maybe	False
Maybe	True	Maybe	Maybe
False	True	True	True

a *is equivalent to* b

≡	True	Maybe	False
True	True	Maybe	False
Maybe	Maybe	Maybe	Maybe
False	False	Maybe	True

Fig. 2. Unknown State Logic Truth Tables

definitionally means that if any input is false, that the output will also be false. This logic is not suddenly invalidated by unknown state logic, instead, the only additional rules come from maybe state: true and maybe is maybe, maybe and maybe is maybe. This is best explained with a truth table, as can be seen in Figure 4 from the Rosetta Code page on Ternary Logic [3]. It is worth noting that this example was given for its similarity to binary logic. Much of ternary logic is less familiar, with many new fundamental operations being available due to the third state.

Just like how binary is not always used for logical operations, but instead as a means of representing numbers in general, the same is true of ternary. The key difference is in the base: As a binary system with 8 digits would only be able to represent unsigned numbers up to 2^8 or 256, a ternary system with as many bits could represent numbers up to 3^8 or 6561. This difference is only made more extreme as the number of digits allotted is increased. One way to visualize this difference is to think of each digit as requiring a separate wire. This brings us to the realization that for a given number, ternary can hold or carry the data using fewer wires. Unfortunately, it also happens that the complexity of ternary also requires more transistors per logic gate. This is best summed up in *The Ternary Manifesto* from Douglas Jones, where he states "The net result is that a ternary computer will generally require on the order of 1.62 times as much logic in its adder as is required by a conventional binary computer of comparable capacity. [...] so it may be that the factor of 1.58 reduction in interconnection repays the cost of the extra gates required for use of Ternary." [2]

II. Brief History of MVL Computers

Multiple Valued Logic (MVL) computation is not a new idea, but to cover this topic in full, one must look back further into computer history, into the era of the analog computer. Analog computers are arguably the most extreme form of MVL, as instead of having a finite number of discrete states, the system operates on continuous signals, doing math in a much more tangible sense. Much like how ternary can have operators that stray from the classic "AND"s and "OR"s of binary, analog computers have historically operated with a different set of basic operations, most notably including integration as a core operation, not something that requires the chaining of other instructions. While analog computers may sound advantageous in this respect, they were vulnerable to noise and required finely tuned components compared to their digital counterparts. Still, analog computers have found some modern success, such as working as low precision neural network acceleration hardware [4].

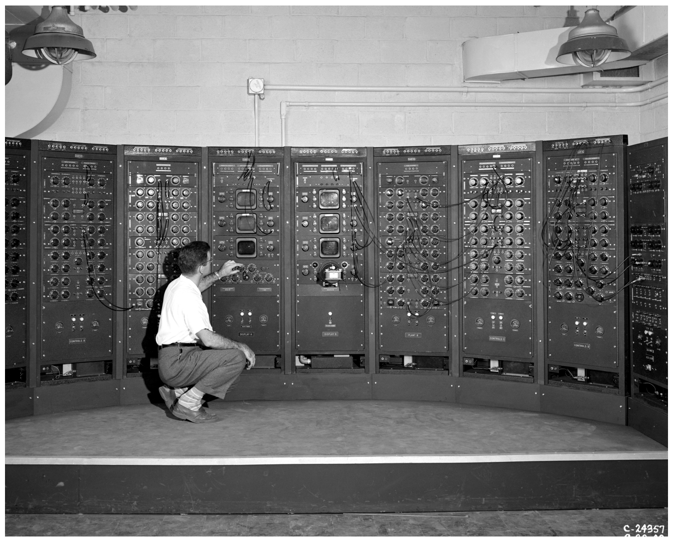
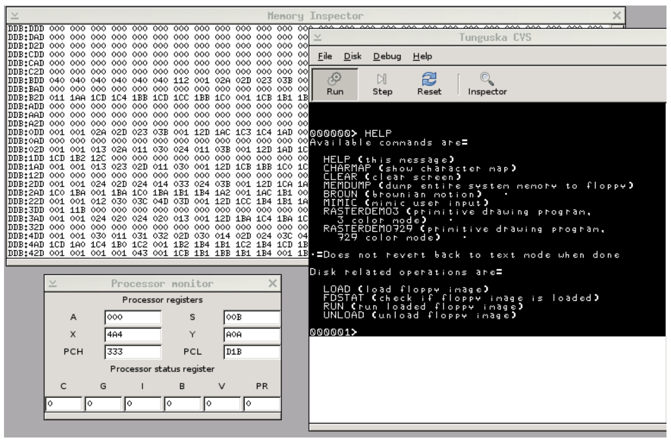


Fig. 3. Analog Computer located in the Engine Research Building at the Lewis Flight Propulsion Laboratory, now John H. Glenn Research Center, Cleveland Ohio.[5]

The mention of analog computers is not just for historical context though. Ternary systems have an inherit analog component. For example, imaging that a ternary system which is not based on electricity, but rather fluids in buckets. The three states may be empty, half full, and totally full. If we simply pour two half full buckets into an empty bucket, we are performing addition. This same process is possible with electricity and a capacitor in an electrical system. This simplistic version ignores the possibility of overflow (quite literally, if you imagine pouring two full buckets into an empty bucket) but the point is that analog logic can be used in a ternary system to simplify and accelerate certain operations. This fact makes comparing different ternary logic implementations complicated. For example, above, it was stated that a ternary system may require about 1.62 times as much



```
DDB:DDD 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:DAD 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:D2D 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:CDD 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:CAD 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:C2D 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:BDD 040 040 040 040 040 040 112 001 02A 02D 023 03B 00
DDB:BAD 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:B2D 011 1AA 1CD 1C4 1BB 1CD 1CC 1BB 1C0 001 1CB 1B1 1B
DDB:ADD 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:AAD 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:A2D 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:0DD 001 001 02A 02D 023 03B 001 12D 1AC 1C3 1C4 1AD 00
DDB:0AD 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:02D 001 001 013 02A 011 030 024 011 03B 001 12D 1AD 1C
DDB:1DD 1CD 1B2 12C 000 000 000 000 000 000 000 000 000 00
DDB:1AD 001 001 013 023 02D 011 030 001 12D 1CB 1BB 1C0 1C
DDB:12D 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:2DD 001 001 024 02D 024 014 033 024 03B 001 12D 1CA 1A
DDB:2AD 1C0 1BA 001 1BA 1C0 1BA 1B1 1B4 1A2 001 1AC 1B1 00
DDB:22D 001 001 012 030 03C 04D 03D 001 12D 1CC 1B4 1B1 1A
DDB:3DD 001 11B 000 000 000 000 000 000 000 000 000 000 00
DDB:3AD 001 001 024 020 024 020 013 001 12D 1BA 1C4 1BA 1C
DDB:32D 000 000 000 000 000 000 000 000 000 000 000 000 00
DDB:4DD 001 001 030 011 031 032 02D 030 014 02D 024 03C 04
DDB:4AD 1CD 1A0 1C4 1B0 1C2 001 1B2 1B4 1B1 1C2 1B4 1CD 1B
DDB:42D 001 001 001 001 043 001 1CB 1B1 1BB 1B1 1B4 001 1B
```

```
000000> HELP
Available commands are=

  HELP (this message)
  CHARMAP (show character map)
  CLEAR (clear screen)
  MEMDUMP (dump entire system memory to floppy)
  BROWN (brownian motion)    ·
  MIMIC (mimic user input)
  RASTERDEMO3 (primitive drawing program,
    3 color mode)    ·
  RASTERDEMO729 (primitive drawing program,
    729 color mode)    ·

·=Does not revert back to text mode when done

Disk related operations are=

  LOAD (load floppy image)
  FDSTAT (check if floppy image is loaded)
  RUN (run loaded floppy image)
  UNLOAD (unload floppy image)

000001>
```

Processor registers

A	000	S	00B
X	4A4	Y	A0A
PCH	333	PCL	D1B

Processor status register

C	G	I	B	V	PR
0	0	0	0	0	0

Fig. 4. Screenshot of the Tunguska emulator [6]

logic as a binary one, but that number assumed no analog circuitry.

As for ternary logic implementations, the history is actually rather sparse. What appears to be the most often referenced ternary system is the Setun from 1959, which used balanced ternary logic. Unfortunately, ternary on the Setun was only really an after-the-fact design decision, as *Ternary Computing Testbed 3-Trit Computer Architecture* points out: "However, its trit flip-flap-flops where not genuine, rather two bits wired to have three stable states." [1]. It is also hard to compare such an old system to any modern hardware, as the Setun comes from an era where computers were still built with vacuum tubes and germanium diodes.

There also does not appear to be any other well documented hardware implementations of ternary systems. Researching online did reveal mentions of other projects, but all of them seem to be lacking anything but a name or maybe a passing sentence in reference. There is, however, at least one notable ternary emulator. The "Tunguska" project is a software emulator with a balanced ternary architecture. It claims to achieve "[...] peak speeds around 1,000,000 operations per second, on a more normal day roughly 250,000 ops/sec" [6]. This project has not gotten an update since 2009 though, so may be capable of better speeds on modern hardware. The project does include a full operating system, assembler, and C compiler as well as a variety of demonstration programs which does point to the project reaching some degree of maturity.

III. Problems Facing Ternary Systems

From the above, it has been established that a ternary architecture is feasible to design, and while examples are scarce, there at least have been some historical examples of such. This begs the question: why do we only see binary logic today?

input	*false*	*identity*	*not*	*true*
false	*false*	*false*	*true*	*true*
true	*false*	*true*	*false*	*true*

Fig. 5. Binary unary operators [2]

input	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*K*	*M*	*N*	*P*	*R*	*T*	*V*	*X*	*Z*
–	–	0	+	–	0	+	–	0	+	–	0	+	–	0	+	–	0	+	–	0	+	–	0	+	–	0	+
0	–	–	–	0	0	0	+	+	+	–	–	–	0	0	0	+	+	+	–	–	–	0	0	0	+	+	+
+	–	–	–	–	–	–	–	–	–	0	0	0	0	0	0	0	0	0	+	+	+	+	+	+	+	+	+

Fig. 6. Ternary unary operators [2]

This comes down to a variety of problems. The most obvious of which is not a technical hurdle but rather a human one: ternary logic is harder to understand. While the simple explanation of unknown state logic and the truth tables shown at the beginning of this paper may make it seem otherwise, ternary logic is inherently much more difficult to mentally work with. To demonstrate this, the basic unary operators of binary and ternary can be examined. In binary, there are only four unary operators. These are the most basic possible logic operations that can be performed. They are done not with two inputs like the fundamental gates (such as "AND" or "OR") but with only one. For binary, there are four (2^2) of these operators and they are limited to forcing the input to be either state, the identity function, and the not function as can be seen in Figure 5. In ternary, there are 27 (3^3) as can be seen in Figure 6 [2]. This issue is only exasperated by going to multiple input logic gates.

Aside from complexity, ternary logic also imposes extra engineering challenges. In binary logic, the two states are typically represented by either a high or low logic signal which correspond to some positive voltage and ground. With ternary, there is a design decision to be made here. One option would be to double the voltage range and have the states balanced around ground (not to be confused with balanced ternary, as any system could be represented with any electrical implementation) such that the states are at -5V, ground, and +5V, for example. Another option is to implement some threshold voltage in the existing range, for example, using ground, +2.5V, and +5V for the three states. Regardless of the choice, the system has gained some electrical complexity and will depend on at least one more stable reference voltage.

Possibly the biggest technical hurdle though is related to this threshold voltage problem. With more states in the same voltage range, it is easier for the system to confuse two values. Referencing the above example where 5V means true, -5V means maybe, and ground means false, there is a problem where confusion may occur if the system tries to read states to quickly and catches a value mid-transition. Should 2V be treated as a true or a maybe? This value confusion becomes more likely as clock speed is increased and the circuit is given less time to reach a steady state. Because of this, historically it has been easier to simply use a binary system and increase the clock speed than it has been to try to implement a ternary system which would likely force the system to use a lower speed.

This at last points to the motivation for this paper. Clock speed and transistor size improvements have been slowing down [7]. This is reflected in the plethora of media outpouring about the ends of Moore's Law. Because Moore's Law is more of an observation than a law, this paper will not comment on the validity of these claims or to what Moore's law actually applies, as the arguments of whether it is about performance or the raw transistor count are endless. The core of the matter is that there is industry wide fear that we are approaching the limits of what silicon based transistors have to offer and lack a clear path forward. As such, all options are on the table. Ternary stands as one of those options.

IV. Ternary after Silicon

While full ternary systems such as the Setun may not be available anymore, that does not mean research into ternary systems has stopped outright. As mentioned in the introduction to this paper, there has been a variety of research done with computation platforms that are not silicon based. Furthermore, some research has explored combining ternary elements into binary systems by converting back and forth as necessary. These ideas have not simply sat stagnant from old research either, as both Complementary metal–oxide–semiconductor (CMOS) [8] and carbon nanotube field-effect transistor (CNFET) [9] binary to ternary converters have been developed recently. This may be implemented much like how modern x86 processors do internal conversion from the complex instruction set computer (CISC) instructions to reduced instruction set computer (RISC) instructions internally for the sake of speed with pipelining [10]. The more obvious reason to use ternary in a binary system may be to simply reduce the number of wires needed for transmission at a given data rate on a bus, as is mentioned in *Design of a Multi-digit Binary-to-Ternary Converter Based on CNTFETs* [9].

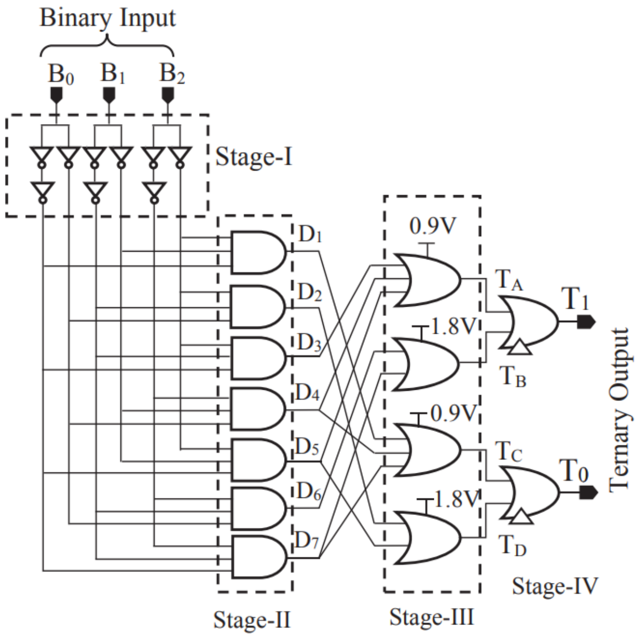


Fig. 7. Binary to Ternary Converter from *DPL-based novel Binary-to-ternary converter on CMOS technology* [8]

A. CNFET and FEFET systems

Carbon nanotube field-effect transistors (CNFET) and ferroelectric transistors (FEFET) have both been used for ternary systems. Both are technologies that aim to replace traditional transistors. In the case of CNFETs, the advantage for ternary comes from the ease of adjusting their threshold voltage. This helps alleviate the previously mentioned problems of confusing states. Some of the papers mentioning ternary logic with CNFETs claim significant improvement, such as *Design of Ternary Logic Circuits using CNTFET* which states "Our design shows approximately 2000× less delay and 10^4x less power as compared to that of existing designs." [11]. Unfortunately, there does not actually appear to be any indication of what 'existing designs' this number is comparing to. There have been some functional implementations done, such as full adders [12] [13].

The FEFETs are seeing success for storing data in a ternary encoding. Again, this is using advantages inherit to the technology. In the case of FEFETs the fact that they can retain a value even without applied voltage provides the advantage. They are being used for doing dot-product computation in memory, as a form of processing in memory (PIM) [14].

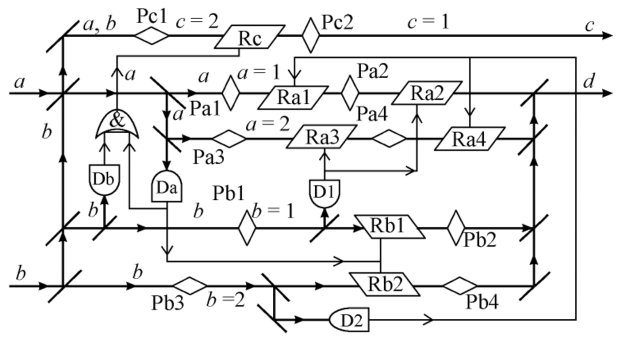


Fig. 8. Optical Ternary Adder [15]

B. Optical Logic

Optical Logic is another area where ternary is being used. Again, the technology lends itself to the ternary system. In this case it is because three states are very easily possible: horizontally polarized, vertically polarized, and off. While explaining the exact mechanism behind optical logic is outside of the scope of this paper, for those familiar with traditional logic gates the system in ternary adder in Figure 8 is at least somewhat familiar, only the logic elements consist of optical elements such as mirrors and polarization films and the lines connecting elements are the paths that a laser may take [15].

This research has already borne some fruit too, as ternary optical logic has been utilized to implement the fast Fourier transform (FFT) with greater efficiency than existing field programmable gate array (FPGA) implementations [16].

Because both ternary systems and optical computers are so foreign to programmers of current systems, there has also been effort put into making their programming and use more approachable via abstraction. In *Programming model and implementation mechanism for ternary optical computer*, Sulan et al. describe a programmer facing programming tool and file format that abstracts away some of the underlying complexity of both ternary and optical computing systems [17].

C. Quantum Computing

Quantum computing is yet another field where ternary has seen some success. Similarly to how the bit is replaced with a trit when changing from the two systems in traditional computing, the qbit is replaced with a qtrit in ternary quantum logic. Intriguingly, there is also research into greater than three level MVL for quantum computing as well, as in *Photonic Qubits, Qutrits and Ququads accurately Prepared and Delivered on Demand* a four level logic system is also outlined [18]. As with the underlying principals of optical logic, trying to explain the fundamentals of quantum computing is out of scope for this paper. What can be said is that quantum computing with MVL has been shown to have some advantages, as in *Multi-Valued Logic Gates for Quantum Computation* the authors state: “We conclude with a comparison of the binary and multi-valued approaches to quantum computing. The main advantage of the latter is a logarithmic reduction in the number of separate quantum systems needed to span the quantum memory” [19].

V. Ternary in Application

A. Machine Learning

Above a few applications were mentioned is passing, such as the FFT algorithm on the optical computer. This section will explore other modern applications of ternary logic.

To begin with, applications around neural networks will be examined. Neural networks, deep learning, and machine learning have all become “buzz words” in the last few years; however, this is not without reason. The field of machine learning has truly been advancing at a rapid rate, both in the way of new ideas and implementations and in the way of sheer scale. For example in advancing to Generative Pre-trained Transformer 3 (GPT3) from its predecessors GPT2 , the model advanced from using to 1.5 billion parameters to 175 billion parameters [20]. Similarly, efforts have been made to scale hardware to meet this higher demand, as is exemplified by the Cerebras “Wafer Scale Engine” machine learning hardware, which, as the name implies, uses processors that are manufactured using the full usable area a silicon wafer, with the latest iteration having 2.6 trillion transistors [21]. Clearly, there is demand for even the most extreme ideas in machine learning acceleration.

It is at this point that some background on the internal workings of a neural network must be discussed. A neural network is a fairly descriptive name, as a given model is made up layers of neurons, often simply called nodes, that are connected to one another. One common network topology involves every node in a layer being connected to every node in the layers directly before and after its own layer. Each of these nodes carries a “weight”. These weights through the network form a sort of path of least resistance that varies depending on the input, to produce a given output. These weights are the values that are learned by giving the model known input and the expected output, until the model hopefully becomes proficient at classifying novel inputs. To better this process each node uses an activation function. The purpose of this function is to scale the output of each node such that one node's output does not dominate or go to zero in way that results in that node being "dead".

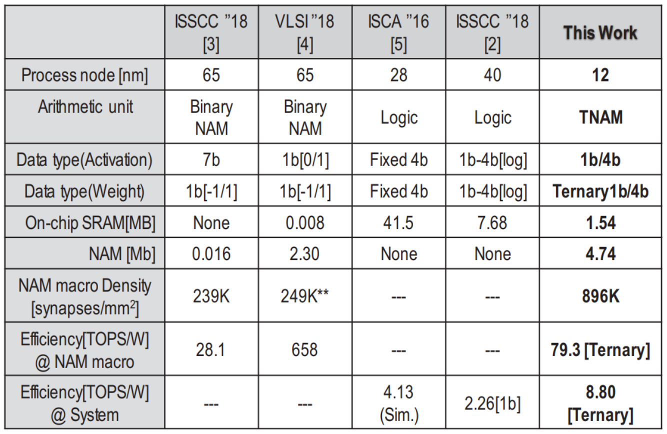
This activation function is where ternary is often applied to machine learning models. Instead of having the weights that exist continuously in the range of the activation function (for example, allowing for a weight of 0.5 for an activation function that limits output to be between 0 and 1), the weights can be quantized to ternary values, only allowing for three values. This does mean that the network loses some information; however, in *HitNet: Hybrid Ternary Recurrent Neural Network* the authors demonstrate that the cost in accuracy that is imposed from this quantization may be worthwhile for the massive memory savings, as storing only three distinct states instead of the high precision value may result in a 16x or better memory savings [22]. This may be combined with previously mentioned technologies, such as the FEFET memory implementation described above.

B. Processing In Memory

Processing in memory (PIM) is another idea for coping with the demands of modern applications as the industry fights to find ways to cope with the end of Moore's law and the bottlenecks inherit in traditional architectures which separate the bulk memory from the computation, where with large data sets it is common for the costs of moving data to be higher than the cost of the actual computation. PIM address this problem by simply moving all or part of the computation into memory itself [23]. Using a ternary memory architecture for PIM on deep neural network workloads has been demonstrated in *A Ternary Based Bit Scalable, 8.80 TOPS/W CNN accelerator with Many-core Processing-in-memory Architecture with 896K synapses/mm^2*. [24]. The success of this method can be seen in Figure 9.

C. Cryptography

Ternary logic has also proven to be useful in cryptography. In *Ternary Computing to Stengthen Cybersecurity*, the authors discuss how ternary based public key exchange algorithms can be used to establish secure communication between parties, even as those parties continue exchange traditional binary data. In Figure 10 from that same paper, it is also demonstrated that ternary can be used for multi-factor authentication by combining multiple authentication methods (Authentication patterns in the figure) in a way that is stated to both increase



	ISSCC "18 [3]	VLSI "18 [4]	ISCA "16 [5]	ISSCC "18 [2]	**This Work**
Process node [nm]	65	65	28	40	**12**
Arithmetic unit	Binary NAM	Binary NAM	Logic	Logic	**TNAM**
Data type(Activation)	7b	1b[0/1]	Fixed 4b	1b-4b[log]	**1b/4b**
Data type(Weight)	1b[-1/1]	1b[-1/1]	Fixed 4b	1b-4b[log]	**Ternary1b/4b**
On-chip SRAM[MB]	None	0.008	41.5	7.68	**1.54**
NAM [Mb]	0.016	2.30	None	None	**4.74**
NAM macro Density [synapses/mm^2]	239K	249K**	---	---	**896K**
Efficiency[TOPS/W] @ NAM macro	28.1	658	---	---	**79.3 [Ternary]**
Efficiency[TOPS/W] @ System	---	---	4.13 (Sim.)	2.26[1b]	**8.80 [Ternary]**

Fig. 9. Results of ternary PIM for CNN acceleration [24]

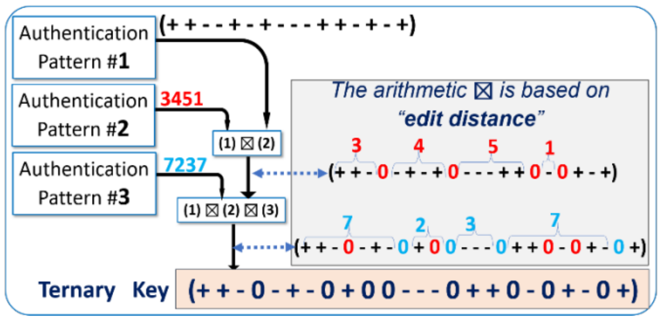


Fig. 10. "Generation of giant ternary keys from multiple authentication patte" [25]

entropy and "reduces the efficiency of sequential attacks" [25].

VI. Conclusion

This paper first provided an overview and history of ternary logic. These sections served to provide context on the difficulty of designing with ternary and how this difficultly in understanding, technical hurdles, and easier paths forward have left ternary logic systems as a less appealing solution; however, the difficultly in advancing via the now traditional means has made it necessary to reevaluate these otherwise overlooked options.

Then, the use of ternary in CNFET, FEFET, Optical, and Quantum based computing media was examined. It was shown that in each, the physics of the media allowed for ternary implementations in a way that was more befitting than ternary would be in traditional silicon FET based architectures, making it a viable option. In Optical computing, references were provided to multiple studies which had made strong proof of concept implementations of adders, an FFT, and a programming method

[15] [16] [17]. Quantum computing with Quitrits and Ququads was also mentioned.

Finally, various applications taking advantage of ternary logic were studied. It was demonstrated that the fields of machine learning, processing in memory, and cryptography have all benefited from recent research into the benefits that using ternary logic for specific problems in each may provide. Machine learning in particular was focused on, as the processing in memory advancements stand to further performance in that area specifically.

This paper aimed to serve as overview of ternary logic, implementations, advancements, and applications. It is the authors personal view that while the advancements in each of the modern hardware options and applications seem quite beneficial, the higher complexity of these systems and the difficultly in understand that comes from it makes ternary of questionable value in all but the most extreme cases. This is not to dismiss the contribution of the research cited. Rather, it is to say while the technical benefit is real and quantifiable, the human cost in engineering time and likelihood of introducing mistakes was not presented in an equally quantifiable way. As such, it is difficult to truly state weather or not ternary may serve as a future option for advancing computer performance as Moore's law becomes a worse predictor of technological advancement.

References

[1] J. Connelly, “Cpe report - ternary computing testbed,” Aug 2008. [Online]. Available: http://xyzzy.freeshell.org/trinary/CPE Report - Ternary Computing Testbed - RC6a.pdf

[2] D. W. Jones, “The ternary manifesto,” Apr 2012. [Online]. Available: http://homepage.divms.uiowa.edu/ jones/ternary/

[3] “Ternary logic.” [Online]. Available: https://rosettacode.org/wiki/Ternary_logic

[4] S. Garg, J. Lou, A. Jain, and M. Nahmias, “Dynamic precision analog computing for neural networks,” 2021.

[5] “Analog computing machine.” [Online]. Available: https://en.wikipedia.org/wiki/File:Analog_Computing_Machine_GPN-2000-000354.jpg

[6] V. Lofgren, “Tunguska,” 2008. [Online]. Available: http://tunguska.sourceforge.net/about.html

[7] T. N. Theis and H.-S. P. Wong, “The end of moore’s law: A new beginning for information technology,” *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[8] A. Saha and D. Pal, “Dpl-based novel binary-to-ternary converter on cmos technology,” May 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1434841117325396

[9] M. Shahangian, S. Hosseini, and S. Komleh, “Design of a multi-digit binary-to-ternary converter based ...” [Online]. Available: https://link.springer.com/article/10.1007/s00034-018-0977-3

[10] J. Sladek and J. R. C. Patterson, “Modern microprocessors a 90-minute guide!” [Online]. Available: http://www.lighterra.com/papers/modernmicroprocessors/

[11] D. Das, A. Banerjee, and V. Prasad, “Design of ternary logic circuits using cntfet,” *2018 International Symposium on Devices, Circuits and Systems (ISDCS)*, 2018.

[12] M. Toulabinejad, M. Taheri, K. Navi, and N. Bagherzadeh, “Toward efficient implementation of basic balanced ternary arithmetic operations in cnfet technology,” *Microelectronics Journal*, vol. 90, p. 267–277, 2019.

[13] S. Firouzi, S. Tabrizchi, F. Sharifi, and A.-H. Badawy, “High performance, variation-tolerant cnfet ternary full adder a process, voltage, and temperature variation-resilient design,” *Computers Electrical Engineering*, vol. 77, p. 205–216, 2019.

[14] S. K. Thirumala, S. Jain, S. K. Gupta, and A. Raghunathan, “Ternary compute-enabled memory using ferroelectric transistors for accelerating deep neural networks,” *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2020.

[15] Y. Jin, “Ternary optical computer principle,” *Science in China Series F*, vol. 46, no. 2, p. 145, 2003.

[16] X. Wei, Y. Shen, J. Peng, Y. Fu, and X. Zhang, “Implementation of parallel fft algorithm on a ternary optical computer,” *SCIENTIA SINICA Informationis*, vol. 47, no. 7, p. 846, 2017.

[17] S. Zhang, J. Peng, Y. Shen, and X. Wang, “Programming model and implementation mechanism for ternary optical computer,” *Optics Communications*, vol. 428, pp. 26–34, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030401818306308

[18] P. B. R. Nisbet-Jones, J. Dilley, A. Holleczek, O. Barter, and A. Kuhn, “Photonic qubits, qutrits and ququads accurately prepared and delivered on demand,” *New Journal of Physics*, vol. 15, no. 5, p. 053007, 2013.

[19] A. Muthukrishnan and C. R. Stroud, “Multi-valued logic gates for quantum computation,” Oct 2018.

[20] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” 2020.

[21] P. Alcorn, “Cerebras second-gen wafer scale chip: 2.6 trillion 7nm transistors, 850,000 cores, 15kw of power,” Apr 2021. [Online]. Available: https://www.tomshardware.com/news/cerebras-wafer-scale-engine-2-worlds-largest-chip-7nm-850000-cores: :text=News-,Cerebras Second-Gen Wafer Scale Chip: 2.6 Trillion 7nm Transistors,850,000 Cores, 15kW of Powertext=The new WSE-2 is,850,000 cores at its disposal.

[22] P. Wang, X. Xie, L. Deng, G. Li, D. Wang, and Y. Xie, “Hitnet: Hybrid ternary recurrent neural network,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS’18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 602–612.

[23] S. Ghose, A. Boroumand, J. S. Kim, J. Gomez-Luna, and O. Mutlu, “Processing-in-memory: A workload-driven perspective,” *IBM Journal of Research and Development*, vol. 63, no. 6, 2019.

[24] S. Okumura, M. Yabuuchi, K. Hijioka, and K. Nose, “A ternary based bit scalable, 8.80 tops/w cnn accelerator with many-core processing-in-memory architecture with 896k synapses/mm2,” *2019 Symposium on VLSI Technology*, 2019.

[25] B. Cambou, P. Flikkema, J. Palmer, D. Telesca, and C. Philabaum, “Can ternary computing improve information assurance?” *Cryptography*, vol. 2, no. 1, p. 6, 2018.

VII. Author's Note

I chose this topic not fully aware of the complexities that lie ahead. While I had a vague idea of the limitations of ternary systems and knew some of the history of analog computers, I did not foresee this topic being as complicated as it was. Reading the literature, old and new, feels as if I'm falling into some alternate reality where people are born with nine fingers instead of ten and so using a three state logic system seems perfectly natural, yet I was still raised with ten fingers and toes, and learned binary for logic. It's all together strange in a way that I don't think simply reading this paper can convey. While I may be wrong, I also suspect that I am not alone in this feeling, as even the author one of the more in depth sources referenced acknowledges his work began as "something of a joke"[2] and stumbling though the corners of the internet and wading my way though less-than-academic sources ternary logic seems to have a bit of a cult following.

My interest in the topic is actually from a bit of an arcane source to begin with: modular synthesis. These are all quite far from the topic of the class for which this paper was written; however, I think there is value to be had in this cross pollination of ideas, and as such I felt this paper incomplete without sharing it.

Modular synthesizers are not all that different from analog computers as mentioned in this paper. In fact, the most popular module for the most popular modular synthesis format, the Make Noise "Maths" for Eurorack, bills itself as an analog computer. Even in a larger eurorack system the modules are patched together and a form of logic, albeit for the purposes of creating music instead of computing differential equations, is performed. Modular synthesis isn't limited strictly to music either. Getting even closer to the realm of analog computing systems, video synthesizers - such as that done by the "Scanimate" machine, which was famously used for some of the graphics in StarWars - and the computing machines used before the dominance of digital are essentially direct relatives.

So, what does this have to do with ternary?

In many synthesizers, Pulse Width Modulation (PWM) is used not for it's traditional engineering use of acting as a cheap way to get a pseudo-analog signal out of a system, but instead as a way of generating more musically interesting and dynamic waveforms. Some of these waves are the traditional two-level variety, but others are actually done with three levels. While this is also something that exists in the realm of engineering (primarily for driving motors), I first saw it on analog oscillators and distortion effects. In some cases, the PWM is actually faked, as instead of actually modulating the pulse width of a generated square wave, instead a comparator (or a pair of them, in the case of the tri-state PWM) is applied to an incoming signal. In most oscillators, this signal is a simple ramp, when crossing the threshold, the value gets quantized. If there are two comparators, there are three quantization states, creating said tri-state wavefrom. First seeing this waveform, which is exactly what would be indicative of ternary, coming out of my modular synth, a device that constantly reminds the user of the analog computing history behind it, sparked this curiosity. I decided that this was an itch worth scratching in full, so that is what I have done with this paper.

It seems odd to admit that the topic for a paper in a subject as rigidly academic as computer architecture is would be inspired from such a strange place; however, I think that the greatest innovations come from the combination of the rigorously defined and the artistic or the academic and arcane. In both the world of art and invention, the most brilliant are also the most insane.