SDPS-2021
Printed in US, December 2021

HIGH SPEED BI-DIRECTIONAL BINARY-TERNARY INTERFACE WITH CNTFETS

Steven Bos, Halvor Nybø Risto, Henning Gundersen
Department of Science and Industry Systems,
University of South-Eastern Norway
steven.bos@usn.no halvor.n.risto@usn.no henning.gundersen@usn.no

ABSTRACT

The world is built on binary electronics, thus high speed and bi-directional radix conversion is needed to enable billions of binary devices to co-exist with non-binary ones. In this article we discuss a generic method for conversion between binary and ternary. A CNTFET implementation is given using a balanced ternary full adder. The implementation is simulated using HSPICE 2020 and is made open source. We demonstrate that nibble word conversion speeds of over 25 GB/s with power consumption of 97.8 μW are achievable with 1760 CNTFETs switching at 5 GHz.

INTRODUCTION

Computers work with binary signals and Boolean logic. Despite analog computers preceding them, discrete signal computers quickly became the standard. In the 1930's John Atanasoff, inventor of the first binary electronic computer, argued for a binary base. In his quest for a faster and more precise computer he realized the hardware to store and process symbols must be low cost, simple and compatible with the rest of the system (Atanasoff, 1940). In that landmark paper he proposed the building blocks of modern computers: capacitors as memory devices, triodes (transistors) as compute devices and implicitly the usage of Boolean algebra.

Modern memory devices no longer store just 2 states. Off-the-shelf solid state drives can have multiple bits stored in a single capacitor, driving down cost, energy consumption and increasing performance and information density (Gulak, 1998). Modern compute devices still use 2-state (binary) signals and Boolean logic. Multiple Valued Logic (MVL) or post-binary alternatives are being actively researched such as ternary CMOS (Jeong et al., 2019), memristor-transistors Zahoor et al., 2021), CNTFETs (Kim, Lim and Kang, 2018) as we approach the physical limits of computing with binary signals (Markov, 2014). Special interest is in computing with ternary signals as they are the closest discrete base to the optimum (Hurst, 1984). Higher bases are still interesting, but the highest average gain for processing large numbers (e.g. 16 bit or higher architectures) can be found when moving from base 2 to 3, which is $log\ 3 / log\ 2 \approx 58.5\%$.

The transition to MVL computers with increasingly higher bases is not instantaneous and requires interfacing with billions of existing (binary) devices. In this paper we focus on machine-machine interfacing at the signal level.

RELATED WORK

Limited work has been published on radix conversion, especially on binary to ternary and inverse converters with focus on circuit implementation. In (Arjmand et al., 2012) an unsigned ternary to unsigned binary converter for Quantum-dot Cellular Automata (QCA) is proposed. While this could generally be implemented in any ternary logic circuit, this is not a capacity efficient conversion method, as two bits are used to store one trit. In (Shahangian, Hosseini and Komleh, 2019), an algorithm for unsigned binary to unbalanced ternary conversion is proposed, with a simulated circuit implementation using CNTFETs. This method is efficient and scalable. In (Li, Morisue and Ogata, 1995) an unsigned binary to balanced ternary converter based on Josephson junctions was proposed. This specific circuit requires superconductors at low temperatures. They use a digit-relation matrix method. We choose to use this method and implement it using CNTFETS and balanced instead of unbalanced ternary. Balanced ternary encoding allows us to discard the sign bit, a source of complexity in both chip design and understanding. We are also aware of the work of Ashur Rafiev (Rafiev, 2011) using Reed-Muller expansions to synthesize unsigned binary to unbalanced ternary radix conversion circuits. This work uses binary CMOS technology for the hardware mapping and focuses on power balancing. Higher radix encoded signals are created with binary signals and is therefore suboptimal. Similarly, in (Iguchi, Sasao, Matsuura, 2006) binary coded ternary is used. The paper presents a generic method implementable in (binary) FPGA and focuses on unbalanced ternary, while we are interested in native ternary and balanced radix conversion.

PARALLEL ARITHMETIC CONVERSION METHOD

Digital computers use a positional numeral system. The value of number N follows from the value of a digit multiplied by a factor determined by the position of the digit. For example in the decimal system the 10 allowable digit values (known as symbols) are 0..9 while each position is a factor of 10 position. The number 16 can thus also be written as $1*101 + 6*100$. We can trivially convert this to another radix such as binary. For example, $2^4 = 16$ and is encoded in binary as 10000_2 or $1*2^4 + 0*2^3 + 0*2^2 + 0*2^1 + 0*2^0$ which will yield the same number. The astute reader will recognize that radix conversion can thus be seen as solving a system of linear equations. The system is a special case of the mixed

radix representation with uniform radix. For binary to ternary these relations until 15-bits and 10-trits are shown in table 2 and for ternary to binary these relations are shown in table 3. Extension to higher order digits is straightforward and reuses all previously found unique rows.

By transposing the digit relation matrix and reverse ordering the rows and columns, a direct conversion matrix appears. Each row now expresses the output at position *i* in base *a* on the left as a sum of the inputs of base *b*. A blank cell has no contributing input but might get input from the carry signal right above it. Any term multiplied with *0* has also no contributing signal. The carry signal corrects the output in case the sum is larger than the number of symbols in that base. Note that there can be situations where the carry signal propagating down to the next row can take up more than one bit or trit and might thus continue to propagate down. This depends on the size of the conversion matrix. The sum, carry and output signals for range [-3,4] can be found in table 1.

Remarkably, no additional functionality is needed for converting negative numbers expressed in balanced ternary into 2's complement notation. For example, inputting the decimal number 5 in ternary $+--_3$ in table 5 results in a carry signal that is added to the sum of each row making the final encoding 101_2 omitting preceding 0's. Inputting the decimal number -5 $-++_3$ into table 5 results in 1111111111111011_2 as the carry will propagate all the way to the sign bit and render the encoding immediately in 2's complement form. Converting from a 2's complement signed binary to a balanced ternary is done by adding extra circuitry when the sign bit is high. This circuitry inverts the input and adds 1. The conversion process is the same as with unbalanced input except that at the end the ternary output is inverted when the sign bit is high.

Table 1 Carry and output signals for binary-ternary converter

	r=2 to r=3		r=3 to r=2	
Sum	Carry	Output	Carry	Output
-3	-1	0	-2	1
-2	-1	+	-1	0
-1	0	-	-1	1
0	0	0	0	0
1	0	+	0	1
2	1	-	1	0
3	1	0	1	1
4	2	+	2	0

4-BIT TO 4-TRIT CNTFET IMPLEMENTATION

A naive implementation of the 15-bit binary to ternary conversion matrix in table 4 with logic gates can be made using only balanced ternary full adders (TFA). In fig. 1 we limited the rows and columns to 4, resulting in a 4 unsigned bit to 4 balanced trit radix converter implementation. A 118

Table 2 Relation between binary and ternary digits

	t_9	t_8	t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0
$2^0 =$										$+3^0$
$2^1 =$									$+3^1$	-3^0
$2^2 =$									$+3^1$	$+3^0$
$2^3 =$								$+3^2$		-3^0
$2^4 =$							$+3^3$	-3^2	-3^1	$+3^0$
$2^5 =$							$+3^3$	$+3^2$	-3^1	-3^0
$2^6 =$						$+3^4$	-3^3	$+3^2$		$+3^0$
$2^7 =$					$+3^5$	-3^4	-3^3	-3^2	$+3^1$	-3^0
$2^8 =$					$+3^5$			$+3^2$	$+3^1$	$+3^0$
$2^9 =$				$+3^6$	-3^5		$+3^3$			-3^0
$2^{10} =$				$+3^6$	$+3^5$	$+3^4$	-3^3		-3^1	$+3^0$
$2^{11} =$			$+3^7$		-3^5	$+3^4$	$+3^3$		-3^1	-3^0
$2^{12} =$		$+3^8$	-3^7		-3^5		-3^3	-3^2		$+3^0$
$2^{13} =$		$+3^8$	$+3^7$	-3^6	$+3^5$	-3^4		$+3^2$	$+3^1$	-3^0
$2^{14} =$	$+3^9$	-3^8	$+3^7$	$+3^6$	$+3^5$	$+3^4$	$+3^3$	-3^2	$+3^1$	$+3^0$

Table 3 Relation between ternary and binary digits

	b^{14}	b^{13}	b^{12}	b^{11}	b^{10}	b^9	b^8	b^7	b^6	b^5	b^4	b^3	b^2	b^1	b^0
$3^0 =$															$+2^0$
$3^1 =$														$+2^1$	$+2^0$
$3^2 =$												$+2^3$			$+2^0$
$3^3 =$											$+2^4$	$+2^3$		$+2^1$	$+2^0$
$3^4 =$									$+2^6$		$+2^4$				$+2^0$
$3^5 =$								$+2^7$	$+2^6$	$+2^5$	$+2^4$			$+2^1$	$+2^0$
$3^6 =$						$+2^9$		$+2^7$	$+2^6$		$+2^4$	$+2^3$			$+2^0$
$3^7 =$				$+2^{11}$				$+2^7$				$+2^3$		$+2^1$	$+2^0$
$3^8 =$			$+2^{12}$	$+2^{11}$			$+2^8$	$+2^7$		$+2^5$					$+2^0$
$3^9 =$	$+2^{14}$			$+2^{11}$	$+2^{10}$			$+2^7$	$+2^6$	$+2^5$				$+2^1$	$+2^0$

transistor TFA made with CNTFETs can be found (Risto, Bos, Gundersen, 2020). Note that for all binary input terms, input circuitry shown in fig. 2 is needed to map a logical 0 in binary to a logical 0 in ternary. This mapping allows negative binary terms and positive binary terms to be fed to the TFA. The naive implementation requires *4 columns*4 rows*0.5 = 6* TFAs for a total of 708 transistors. In addition, input circuitry is needed: *5*6=30* binary level shifters for positive terms (unary function R) and *2*4=8* binary level shifters (unary function R) for negative terms. The total transistor count is thus *708+38=746*. Not counted is the input circuitry for handling 2's complementary signed binary input as that is optional.

Several optimizations are immediately apparent. Empty cells in the matrix indicate that one input signal is not needed and thus the full adder with 3 inputs can be simplified to a ternary half adder (THA) with two inputs, reducing transistor count from 118 to 50 saving 68 transistors per empty cell. The THA can be implemented with CNTFETs using dyadic functions 7PB for the sum and RDC for the carry. Since the entire first row does not require a carry signal from the previous row, all TFA can be replaced for THA saving an additional *3*(118-50)* = 204 transistors, for a total of *272/746 =36%* with two trivial optimizations.

Another 30 to 40% can be saved by optimizing the logic gates individually by inspecting its input signals and output signal requirements and assign don't care states. For example, binary input signals require simpler ternary adders. Another example is that the carry signal is often quite limited in range. This also means that we can integrate input circuitry into the adders directly.

Table 4 15-bit to 10-trit conversion matrix

t_0 =	+ b_{14}	- b_{13}	+ b_{12}	- b_{11}	+ b_{10}	- b_9	+ b_8	- b_7	+ b_6	- b_5	+ b_4	- b_3	+ b_2	- b_1	+ b_0
t_1 =	+ b_{14}	+ b_{13}		- b_{11}	- b_{10}		+ b_8	+ b_7		- b_5	- b_4		+ b_2	+ b_1	
t_2 =	- b_{14}	+ b_{13}	- b_{12}				+ b_8	- b_7	+ b_6	+ b_5	- b_4	+ b_3			
t_3 =	+ b_{14}		- b_{12}	+ b_{11}	- b_{10}	+ b_9		- b_7	- b_6	+ b_5	+ b_4				
t_4 =	+ b_{14}	- b_{13}		+ b_{11}	+ b_{10}			- b_7	+ b_6						
t_5 =	+ b_{14}	+ b_{13}	- b_{12}	- b_{11}	+ b_{10}	- b_9	+ b_8	+ b_7							
t_6=	+ b_{14}	- b_{13}			+ b_{10}	+ b_9									
t_7 =	+b_{14}	+ b_{13}	- b_{12}	+ b_{11}											
t_8 =	- b_{14}	+ b_{13}	+ b_{12}												
t_9 =	+b_{14}														

Table 5 10-trit to 15-bit conversion matrix

b_0 =	+ t_9	+ t_8	+ t_7	+ t_6	+ t_5	+ t_4	+ t_3	+ t_2	+ t_1	+ t_0
b_1 =	+ t_9		+ t_7		+ t_5		+ t_3		+ t_1	
b_2 =										
b_3 =			+ t_7	+ t_6			+ t_3	+ t_2		
b_4 =				+ t_6	+ t_5	+ t_4	+ t_3			
b_5 =	+ t_9	+ t_8			+ t_5					
b_6 =	+ t_9			+ t_6	+ t_5	+ t_4				
b_7 =	+ t_9	+ t_8	+ t_7	+ t_6	+ t_5					
b_8 =		+ t_8								
b_9 =				+ t_6						
b_{10}=	+ t_9									
b_{11}=	+ t_9	+ t_8	+ t_7							
b_{12}=		+ t_8								
b_{13}=										
b_{14}=	+ t_9									

We present an early attempt to an optimized unsigned binary to balanced ternary radix converter in Fig. 4. The 6 TFA's used above each had 1 triadic (3 input) carry and 2 dyadic (2 input) sum logic gate, resulting in 6 triadic components, and 12 dyadic components. Our proposed radix converter has 2 triadic components and 9 dyadic components and includes input circuitry such as inverters. The total transistor count is 176, a reduction of 76% compared to the naive approach. All components are adaptions of the sum and carry logic gates. The two indices in the names indicate which parts of the conversion matrix they address e.g. index *00* means row 0, column 0. The postscript a or b denotes the 1st or 2nd sum logic gate in the hybrid TFA. Multiple variants are needed as input terms are sometimes negative or carry does not propagate resulting in various heptavintimal indices.

HSPICE SIMULATION

Fig. 3 shows the HSPICE output of the proposed radix converter from Fig 4. Table 6 shows the performance of the proposed design compared to the state-of-the-art. Note that the comparison is not completely fair as the radix converter in (Shahangian, Hosseini and Komleh, 2019) uses unbalanced ternary which generally requires fewer transistors.

Table 6 Simulation results at 5 GHz input frequency

Radix converter	Transistors	Avg. Power	Worst measured delay
4b4t (unbal.)[1]	151	64.41e-06 W	46.95ps
proposed 4b4t (bal.)	176	9.78e-06 W	80ps

[1] **(Shahangian, Hosseini and Komleh, 2019)**

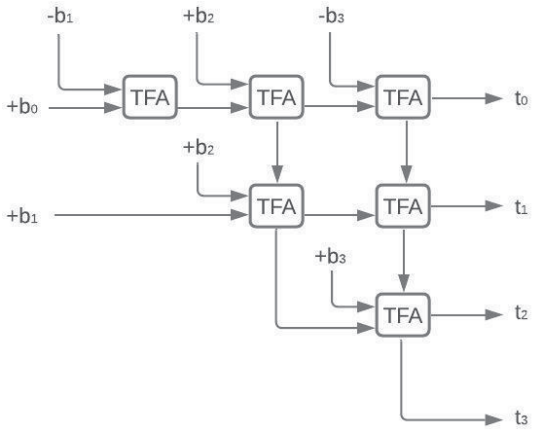

Fig. 1 A naive implementation of the 4-bit to 4-trit conversion matrix found in table 4 using only hybrid Balanced Ternary Full Adders (TFA) from (Risto, Bos, Gundersen, 2020). It covers the first columns and rows from the table

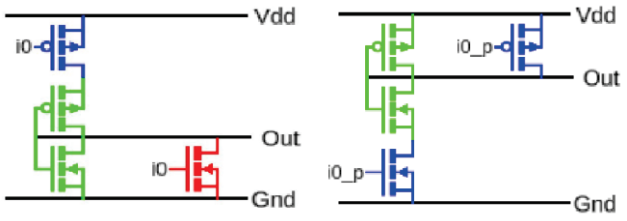

Fig. 2 Input circuitry to make binary signals compatible with balanced ternary. The color of the transistor denotes different CNTFET types with different activation thresholds for the middle value (Kim, Lim and Kang, 2018). *(Left)* The unary function 4 in heptavintimal notation has 4 transistors inverting every binary 1 to a ternary -1 (ground). *(Right)* The unary function R in heptavintimal notation has 6 transistors, 4 are shown. 2 more are needed for a Positive Ternary Inverter (PTI) in front of the input *i0_p*.

DISCUSSION

Adders circuits are central in many functional units such as shift-add circuits and arithmetic logic units (ALU's), so further optimizing them is generally worthwhile. Logic gate design is an art, and many more optimizations can be made when studying radix conversion circuit at the transistor level. Our proposed radix converter is capable of high-speed processing and has low power consumption. It is comparable to the state of the art, with nearly matching transistor count and outperforming it in power consumption by -85%. In addition, the circuit can handle add/subtract arithmetic due to its balanced ternary nature. With little circuitry we can assign the most significant bit (MSB) to be the sign bit and automatically convert signed binary to balanced ternary. This makes the proposed radix converter a multi-purpose radix converter. The inverse, a ternary to binary radix converter can be constructed by using the same procedure but now based on Table 5. An optimized version would use ternary input-binary

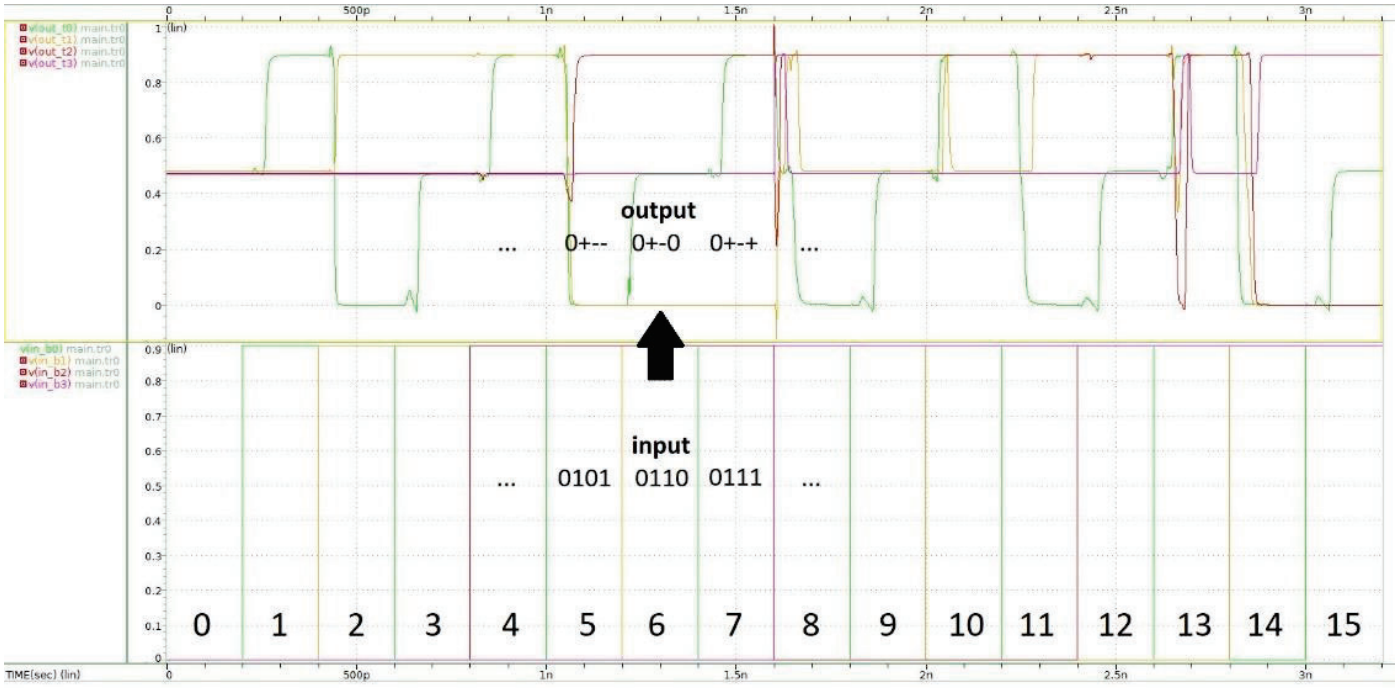

Fig. 3 *(Top)* Transient simulation of trit signals T0..T3. of the proposed 4b4t radix converter in fig. 4. The transition from decimal 7 to 8 determines the worst case delay of 80 ps (12.5 GHz). *(Bottom)* The binary inputs signals B0..B3 follow a 200 ps (5GHz) cadence going from 0000 to 1111 (decimal 0 to decimal 15).

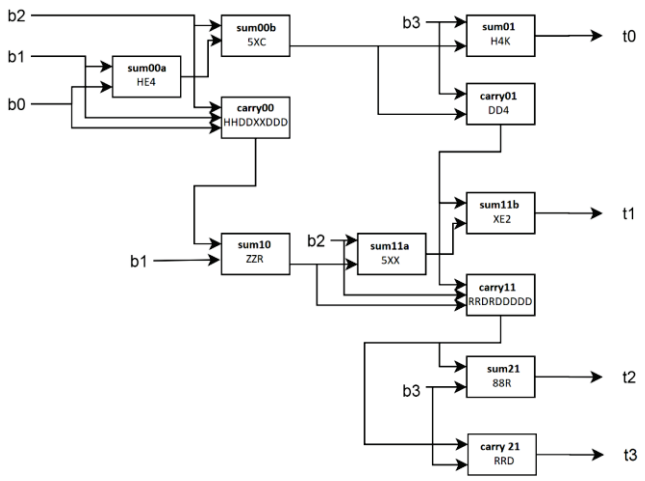

Fig. 4 Block schematic of the proposed unsigned 4 bit to balanced 4 trit radix converter. The 11 CTNFET circuits schematics of the logic gates are available (Bos and Risto, 2021).

output sum logic gates and ternary carry logic gates. In combination with a 1:2 multiplexer a single bi-directional circuit can be constructed where the multiplexer selects which conversion circuit should be active.

CONCLUSION

A generic look-up table (LUT) method to generate and optimize signed binary and balanced ternary radix conversion matrices have been presented. Simulation data and netlist files of the 11 logic gates are made open source (Bos and Risto, 2021). The method can easily be extended to 64 bits or higher. A 4-bit version has been simulated in HSPICE. This circuit can be connected in parallel if we consider words to be 4 bits (one nibble). This means data can be converted at a rate of 9.78 μW and 176 transistors per 2.5 GB/s when the input frequency is 5 GHz. For example a 25GB/s radix converter would have a power consumption of 97.8 μW and component count of 1760 transistors, ideal for data heavy but energy efficient multi-radix IoT devices. Higher frequency operation should be possible as the worst case delay is 80 ps, about 12.5 GHz instead of the current 5 GHz.

FUTURE WORK

As CNTFET circuit fabrication is currently only possibly in highly specialized fabs, research is needed to investigate if radix converters can also be realized with standard CMOS technology (Morozov, Pilipko and Korotkov, 2009). With carefully matched off-the-shelf MOSFETs stable middle voltage levels can be realized. We were able to correctly simulate and prototype this on a breadboard. By using standard CMOS fabrication, price points of these conversion chips can become low, speeding up the transition from pure binary to mixed-radix circuits.

ACKNOWLEDGMENT

We thank the members of the ternary research group for their feedback on this paper.

REFERENCES

Arjmand, M.M., Soryani, M., Navi, K. and Tehrani, M.A. (2012). A novel ternary-to-binary converter in quantum-dot cellular automata. *2012 IEEE Computer Society Annual Symposium on VLSI.* pp. 147–152.

Atanasoff, J.V. (1940). Computing Machine for the Solution of large Systems of Linear Algebraic Equations. *Origins of Digital Computers: Selected Papers, Springer-Verlag, Berlin and Heidelberg, 1982,* pp. 315–336.

Bos, S., and Risto, H.N. (2021). High speed bi-directional binary-ternary interface with CNTFETS. [Computer software] github.com/aiunderstand/MixedRadixConversion

Gulak, P.G. (1998). A review of multiple-valued memory technology. Proceedings 1998 28th IEEE International Symposium on Multiple-Valued-Logic, pp. 222–231.

Hurst, S.L. (1984). Multiple-valued logic - its status and its future. *IEEE Transactions on Computers, 33(12)*. pp. 1160–1179.

Iguchi, Y., Sasao, T. and Matsuura, M. (2006). On designs of radix converters using arithmetic decompositions. *IEEE International Symposium on Multiple-Valued Logic.* pp. 1 – 6.

Jeong, J.W., Choi, Y.-E., Kim, W.-S., Park, J.-H., Kim, S., Shin, S., Lee, K., Chang, J., Kim, S.-J. and Kim, K. (2019). Tunnelling-based ternary metal–oxide–semiconductor technology. *Nature Electronics*, *vol. 2,* pp. 307-312.

Kim, S., Lim, T., and Kang, S. (2018). An optimal gate design for the synthesis of ternary logic circuits. *23rd Asia and South Pacific Design Automation Conference (ASP-DAC).* pp. 476–481.

Li, F-Q., Morisue, M., and Ogata, T. (1995). A proposal of josephson binary-to-ternary converter. *IEEE Transactions on Applied Superconductivity, 5(2)*, pp. 2632–2635.

Markov, I. (2014). Limits on fundamental limits to computation. *Nature, 512.* pp. 147–54.

Morozov, D.V., Pilipko, M.M. and Korotkov, A.S. (2009). The design of ternary logic units on the basis of the standard MOS technology. *Russian Microelectronics, 38(3)*. p. 206.

Rafiev, A. (2011). Mixed radix design flow for security applications. *Ph.D. dissertation, Newcastle University.*

Risto, H.N., Bos, S. and Gundersen, H. (2020). Automated synthesis of netlists for ternary-valued n-ary logic functions in CNTFET circuits. *Linköping University Press in Linköping Electronic Conference Proceedings, 176*. pp. 483–485.

Shahangian, M., Hosseini, S.A., and Pishgar Komleh, S.H. (2019) Design of a multi-digit binary-to-ternary converter based on cntfets. *Circuits, Systems, and Signal Processing, 38(6).* pp. 2544–2563.

Zahoor, F., Hussin, F.A., Khanday, F.A., Ahmad, M.R., Mohd Nawi, I., Ooi, C.Y. and Rokhani, F. Z. (2021). Carbon nanotube field effect transistor (CNTFET) and resistive random access memory (RRAM) based ternary combinational logic circuits. *Electronics, 10(1), 79.*