

High Efficiency Multiply-Accumulator Using Ternary Logic and Ternary Approximate Algorithm

Wanting Wen¹, Guangchao Zhao², Wanbo Hu, Ziyi Li, Xingli Wang³, Eby G. Friedman⁴, *Life Fellow, IEEE*, Beng Kang Tay⁵, *Senior Member, IEEE*, Shaolin Ke⁶, and Mingqiang Huang⁷

Abstract—A multiply-accumulator, often abbreviated as a MAC unit, is central to a multitude of computational tasks, particularly those tasks (such as neural networks) involving array-based mathematical computations. The quest for novel methods to efficiently store and process data in a MAC has become imperative. Recently, ternary logic has attracted significant attention due to its higher information density than conventional binary systems. However, though numerous studies have showcased ternary arithmetic circuits, advancements in ternary-based vector processing have been notably scarce. To bridge this gap, this work undertakes comprehensive study into the optimization of ternary MAC units. Firstly, we propose various ternary approximate algorithms which shows 30%-less power consumption and only 2% computation error when compared with the accurate design. Secondly, we design sophisticated ternary circuits and obtain 74%~80% lower power-delay-product (PDP) than previous works. Finally, we evaluate the proposed ternary MAC unit using both carbon-nanotube field-effect transistor (CNTFET) and silicon-based 180 nm CMOS processes. The simulation results show the ternary circuit is better than binary circuit in terms of both area (~45% less) and power (~30% less), highlighting its strong potential for practical applications.

Index Terms—Ternary logic circuit, multiplying-accumulator, approximation algorithms.

Received 10 July 2024; revised 7 October 2024; accepted 28 October 2024. This work was supported in part by STI 2030, Major Projects under Grant 2022ZD0210600; and in part by the Natural Science Foundation of Guangdong Province under Grant 2023B1515020051. This article was recommended by Associate Editor X. Fong. (Wanting Wen, Guangchao Zhao, and Wanbo Hu contributed equally to this work.) (Corresponding authors: Shaolin Ke; Mingqiang Huang.)

Wanting Wen is with Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, and also with Hubei Key Laboratory of Optical Information and Pattern Recognition, Wuhan Institute of Technology, Wuhan 430205, China.

Guangchao Zhao, Xingli Wang, and Beng Kang Tay are with the Centre for Micro- and Nano-Electronics (CMNE), School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

Wanbo Hu, Ziyi Li, and Mingqiang Huang are with Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: mq.huang2@siat.ac.cn).

Eby G. Friedman is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA.

Shaolin Ke is with Hubei Key Laboratory of Optical Information and Pattern Recognition, Wuhan Institute of Technology, Wuhan 430205, China (e-mail: keshaoлин@wit.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TCSI.2024.3492797>.

Digital Object Identifier 10.1109/TCSI.2024.3492797

I. INTRODUCTION

WITH the widespread use of artificial intelligence (AI), Internet of Things (IoT), and autopilot, there is a rapidly growing demand for processing data [1], [2]. However, with the coming end of Moore's Law, it becomes more difficult to improve the performance of integrated circuits (IC) by shrinking the size of the transistor features [3]. New "more-than-Moore" approach for efficient data storage and computing is highly desired. Multi-valued logic (MVL) systems possess a greater number of logical states as compared with traditional binary logic systems (0, 1), inherently offering higher data density and enhanced computational speed.

In the past few decades, various multi-valued logic computer systems have been realized, such as the ENIAC (with decimal I/O and logic circuits) [4], C₆TYHb-70 (with ternary logic circuits) [5]. In addition to the hardware manufacturability, the key point of which multi-valued logic should be used strongly depends on the parameters related to information technology, such as noise margin and hardware efficiency. Noise margin is a critical metric for evaluating the robustness of digital circuits, and it can be derived from the butterfly curve of the voltage transfer characteristic. Generally, noise margin and information density are two opposing traits, and it is essential to find the optimal trade-off between them (Supplementary Material S1). Besides, among all the integer radix, the radix-3 shows the highest storage efficiency (Supplementary Material S2). This unique characteristic makes ternary logic an attractive choice for applications where data density and hardware efficiency are critical factors.

The development of ternary circuits began in the 1970s. Since then, numerous studies have explored ternary functions using various technologies, including CMOS with resistors [6], memristors [7], 2D semiconductor hetero-junctions [8], [9], quantum dots [10], and negative capacitance devices [11], [12]. However, these studies largely remain fragmented, with only a few simple modules researched, lacking an overall systematic framework. In comparison, carbon nanotube ternary logic is the most extensively researched and well-systematized field. Carbon nanotube field-effect transistor (CNTFET) has been explored for the realization of basic ternary logic gates over the past decade [13], [14], [15], [16], [17], [18], [19], [20], due to its ability to change the threshold voltage (V_{th}) by adjusting

the geometric structure (such as diameter). The relationship between the threshold voltage and the diameter of the CNT (D_{CNT}) can be expressed as $V_{th1}/V_{th2} = D_{CNT2}/D_{CNT1}$. V_{th} is inversely proportional to D_{CNT} , enabling precise control of V_{th} by tuning the D_{CNT} . This capability makes CNTFET a promising platform for ternary logic systems.

Up to now, there have been many ternary circuit implementations based on CNT. For example, Jaber et al. proposed ternary inverters and ternary half adders (THA) based on CNTFET [13], [14]. Firouzi et al. developed a ternary full adder, which exploits earlier topologies to achieve higher performance while maintaining low complexity [15], [16], [17]. Asibelagh et al. described 6-trit ternary multiplier [18]. Srinivasu et al. developed a synthesis technique applicable to devices such as CNTFET that supports ternary logic to provide circuits with low transistor costs [19], [20]. Recently, more complex and efficient ternary logic circuits have been introduced, such as those optimized through algorithmic improvements [21]. For example, Zhao et al. proposed a ternary cycling gate to construct ternary adders [22]. This innovation reduces both the transistor count (area) and energy consumption (power) of ternary adders. [23] proposed a 6-trit \times 6-trit ternary multiplier in the approximate domain. Kim et al. introduced a 16-bit \times 16-bit approximate multiplier by employing low-energy approximate adders for the lower bits [24]. However, their proposed multiplier significantly increased 22.2% area, 20% delay, and 21.5% power, as compared with conventional accurate binary multipliers, indicating further optimizations on the ternary system.

Another key issue of ternary system is the compatibility with existing CMOS technologies. Though previous studies have investigated CNT based ternary functions, attributed to the tunability of the V_{th} and comprehensive logic synthesis techniques for arbitrary ternary logic functions [25], [26]. The CNTFET is indeed incompatible with current manufacturing processes [27], and it is also difficult to control the CNT diameter. On the contrary, by adjusting the doping profile and oxide thickness in CMOS process, the threshold voltage of MOSFETs can be precisely controlled. Consequently, this work explores the development of ternary logic circuits using a commercial CMOS process to address challenges in large-scale fabrication.

This study focuses on the design of an approximate ternary multiplication-accumulation unit, a fundamental building block for both deep neural network applications and current digital signal processing systems. The primary contributions of this research are:

1. We proposed various ternary arithmetic circuits including ternary approximate adders, ternary approximate/accurate 4-2 compressors, and ternary approximate/accurate multipliers. Then we design low-power and minimal-error ternary approximate multiplying accumulator unit by employing the aforementioned ternary cells, all of which have been evaluated using 32 nm CNTFET technology and shows higher performance compared with previous works.
2. We verify the proposed design using a Semiconductor Manufacturing International Corporation (SMIC) 180 nm

silicon process. Our exploration of using commercial CMOS technology to construct proposed ternary circuits helps to address the challenges regarding the incompatibility with the mature manufacturing process and practical fabrication.

The remainder of this article is organized as follows. Section-II describes the background knowledge on ternary representation scheme and the signed and accurate ternary logic circuit. Section-III presents the CNT-based middle-scale approximate ternary logic circuits ($\sim 1,000$ transistors), such as 2-trit multiplier and 2-trit multiply-accumulator. Section-IV presents the CNT based large-scale approximate ternary logic circuits ($\sim 10,000$ transistors), including 6-trit multiplier and 6-trit multiply-accumulator. Section-V shows the performance comparisons of ternary circuit and binary circuit (both using CNT technology) on accelerating one layer of ternary neural network. Section-VI presents the performance comparisons of ternary and binary logic using Silicon technology to demonstrate the manufacturability and CMOS-compatibility. Finally, section-VII summaries the whole work.

II. BACKGROUND AND TERNARY LOGIC SYSTEM

In this section, we discuss the disparity between unbalanced and balanced ternary representation schemes, and then we design accurate and signed ternary multiplier/MAC unit.

A. Balanced and Unbalanced Ternary Representation

Two representations for ternary logic systems exist, the unbalanced ternary logic (UBT: 0, 1, 2) and balanced ternary logic (BT: -1, 0, 1). Similar to the encoding approach of binary 1's complement, we introduce a ternary complement encoding scheme. The most significant digit is used as the signed-trit, 2 (or 1) represents the negative, and 0 represents a positive value. To convert a decimal to an unbalanced ternary n -trit number,

$$Y_n = \pm \sum_{i=0}^{n-2} y_i 3^i$$

The signed-trit of y_{n-1} does not participate in numerical computation. For example, the unbalanced ternary number of $(0101)_3$ equals $(10)_{10}$ in decimal, because $(0101)_3 = 1 \cdot 3^2 + 1 = 10$. The ternary number of $(1101)_3$ is $(-10)_{10}$ in decimal.

Signed digit extension is an important feature for data representation. In binary 2's complement, the binary data can be sign-extended without changing the value, which can be used to accumulate different partial-products in a signed multiplier. Fortunately, the 3's complement ternary representation also exhibits these characteristic. As shown in **Table I**, the decimal value of -10 can be represented by $(2 \ 122)_3$ in 3's complement, and can also be sign-extended as $(22 \ 122)_3$ or $(22222 \ 122)_3$.

For the balanced ternary encoding scheme (-1/0/1), the number itself has natural negative values, so no need to expand the signed-trits. The balanced ternary number can encode positive and negative numbers in Type-3,

$$Y_n = \sum_{i=0}^{n-1} y_i 3^i$$

TABLE I
SIGNED TERNARY REPRESENTATIONS

Decimal	Unbalanced 1's Complement Type-1	Unbalanced 3's Complement Type-2	Balanced Type-3
10	0 101	0 101	101
9	0 100	0 100	100
7	0 021	0 021	1T1
6	0 020	0 020	1T0
3	0 010	0 010	010
2	0 002	0 002	01T
1	0 001	0 001	001
0	0 000	0 000	000
-1	2 001	2 222	00T
-2	2 002	2 221	0T1
-3	2 010	2 220	0T0
-6	2 020	2 210	T10
-7	2 021	2 202	T1T
-9	2 100	2 200	T00
-10	2 101	2 122	T0T
-10 (sign bit extension)	2 00 101	222 122	00 T0T
-10 (sign bit extension)	2 00000101	222222 122	0000 T0T

where y_i is the value of the trit, which can be -1, 0, or 1. The base-3 weighting can be represented in 3^i .

Table I lists the ternary representation for different decimal numbers. Note that the balanced representation takes full advantage of the data range. Unbalanced methods waste several data representations because the signed-trit cannot be 1, which is related to signed-extension and signed arithmetic adders. Note that despite the difference between unbalanced and balanced representations, both can be electrically represented by (0, $V_{DD}/2$, V_{DD}).

B. Accurate and Signed Ternary Multiplier

Fig. 1 shows the computational flow of a signed multiplier in different data representation forms. Since many related works on the design of a 1-trit adder or 1-trit multiplier exist [13], [14], the focus here is on the multi-trit multiplier architecture. The circuit design of ternary half adders, full adders, and multipliers is described in the literature [13], [14], [15], [17], [24].

The structure of a Type-1 signed multiplier is depicted in **Fig. 1(a)**. In the unbalanced 1's complement representation, the most significant digit is the signed-trit and does not participate in the numerical computation process. The multiplier can therefore be constructed by dividing the task into two sub-tasks: one for the signed-trit and the other to compute the mantissa. The signed-trit can be directly deduced from a binary XNOR gate. When the inputs are both positive or both negative, the results are positive and the signed-trit is 0; otherwise, it is negative and the signed-trit is 2. For the mantissa, since both the multiplicand and multiplier

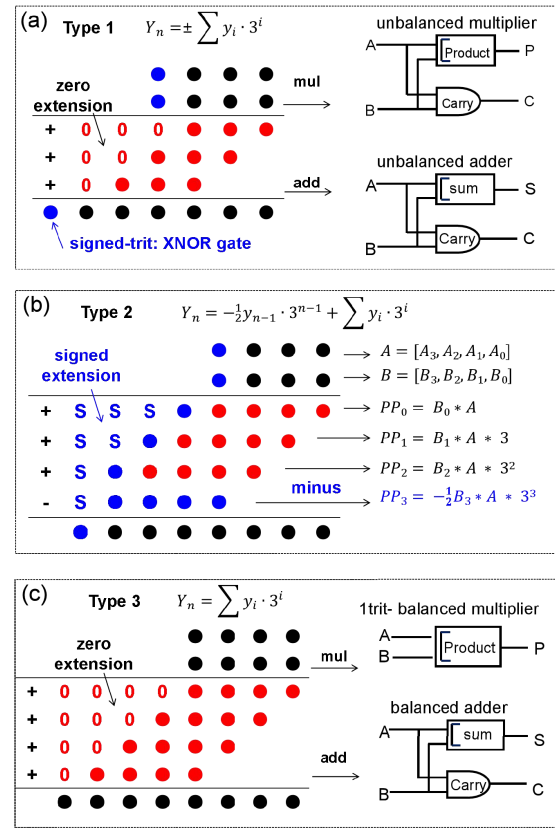


Fig. 1. Computation flow of signed 4-trit \times 4-trit multiplier: (a) Type-1, 1's complement unbalanced ternary; (b) Type-2, 3's complement unbalanced ternary; (c) Type-3, balanced ternary.

are positive, the partial-products are also positive. Therefore, we only need zero-extension during the accumulation stage.

In Type-1 representation, the signed-trit only expresses positive and negative values; thus the overhead of the hardware circuit is rather low. Once the signed-trit participates in the computation (Type-2), the circuit becomes more complicated. In this case, all of the partial-products are summed in the signed-extension scheme. **Fig. 1(b)** exhibits an example 4-trit \times 4-trit multiplier in an unbalanced ternary scheme. We consider B as the multiplicand. Each trit of B multiplies the entire A and generates partial-products of PP_0 , PP_1 , and PP_2 (indicated by the red dots). For the signed-trit of B (i.e., B3), note that the weight of B3 can only be 0 or $-1 \cdot 3^3$ (not $-2 \cdot 3^3$); therefore, the last accumulation operation is actually a subtractor.

Fig. 1(c) illustrates the 4-trit \times 4-trit multiplier in the Type-3 balanced ternary scheme. Due to the uniqueness of the balanced ternary encoding scheme, it does not produce a carry signal during the 1-trit multiplication. The partial product generation part is therefore much more lightweight than a unbalanced multiplier. The subsequent step entails using a THA, TFA, or 4-2 compressor for the final summation. Since the balanced encoding scheme does not require the special signed-trit, the sign-bit extension is also eliminated.

Another important issue is the data range. For the 4-trit signed multiplier, the minimum/maximum input data is -26 to 26 for Type-1, -27 to 26 for Type-2, and -40 to 40 for Type-3.

TABLE II
PERFORMANCE COMPARISON OF THE SIGNED ACCURATE
TERNARY MULTIPLIER (@ 32 nm CNT)

CNT 4-trit MUL	Transistor count	Freq. (GHz)	Delay (ps)	Avg power (uW)	PDP (aJ)	Normalized power
Type-1	1530	1	816.60	5.23	4267.55	98.68
		0.8	824.13	5.06	4169.27	95.47
		0.5	863.12	2.59	2232.81	48.87
		0.4	878.72	1.85	1628.45	34.91
		0.2	991.32	1.09	1081.33	20.57
In-data range [-26,26]	Out-data range [-676,676]	0.1	1211.91	0.99	1200.28	18.68
Type-2	5649	0.8	timing failed	-	-	-
		0.5	1595.38	5.01	7993.81	92.78
		0.4	1623.79	3.28	5323.40	60.74
		0.2	1813.98	2.58	4687.30	47.78
		0.1	2167.67	1.61	3491.99	29.81
In-data range [-27,26]	Out-data range [-702,729]	0.1	2167.67	1.61	3491.99	29.81
Type-3	1584	0.8	timing failed	-	-	-
		0.5	timing failed	-	-	-
		0.4	2269.32	2.60	5903.85	32.10
		0.2	2391.99	1.61	3844.17	19.88
		0.1	2641.99	0.59	1550.58	7.28
In-data range [-40,40]	Out-data range [-1600,1600]	0.1	2641.99	0.59	1550.58	7.28

The minimum/maximum output data is -676 to 676 for Type-1, -702 to 729 for Type-2, and -1600 to 1600 for Type-3.

To validate the performance of the signed ternary multipliers circuit, transient simulations are conducted using HSPICE with the 32 nm CNTFET library [20]. Each test circuit is simulated with random inputs at supply voltage $V_{DD} = 0.9$ V, temperature $T = 27$ °C, and different frequency. **Table II** is a performance comparison of the three multipliers. Whether considering a practical data range, the balanced ternary (Type-3) always consumes less energy and area, and the normalized power consumption (defined as Avg_Power/Data_Range) is only half of the unbalanced ternary logic. The circuit, however, cannot work at a frequency higher than 500 MHz. For the unbalanced ternary scheme, Type-1 exhibits better performance than Type-2.

C. Signed Ternary Multiplying-Accumulator

The typical multiplication-accumulation operation requires both signed multiplication and signed accumulation. Though the Type-1 unbalanced ternary scheme (1's complement) is effective at high-speed computing, it is not suitable for signed accumulation. The Type-2 unbalanced ternary 3's complement executes signed addition and scheme with signed bit extension; thus it can be used in signed ternary multiplication-accumulation. As to the balanced ternary encoding scheme, both the signed multiplication and signed accumulation can be directly executed. To compare the performance of balanced ternary and unbalanced ternary signed computations, we construct several MAC trees with different input vector lengths (M) in HSPICE using the 32 nm CNTFET library [20] and compare the performance at different frequencies in **Table III**.

TABLE III
PERFORMANCE COMPARISON OF THE SIGNED ACCURATE
TERNARY MAC (@ 32 nm CNT)

CNT 4-trit MAC	Transistor count	Vector size	Freq. (GHz)	Delay (ps)	Avg power (uW)	PDP (aJ)	Normalized power
Type-1	Type-1 is unsigned, therefore not suitable for the signed MAC						
Type-2 unbalance	12638	M=2	0.2	3170.43	6.27	19883.45	116.11
			0.1	3772.41	4.08	15383	75.56
	19355	M=3	0.2	3514.82	9.57	33651.82	177.22
			0.1	4174.22	6.24	26037.98	115.56
	26124	M=4	0.2	3801.06	12.91	49058.42	239.07
			0.1	4486.64	8.42	37766.45	155.93
Type-3 balance	4594	M=2	0.2	4816.04	4.55	21914.82	56.17
			0.1	5293.11	2.1	11095.1	25.93
	7332	M=3	0.2	5459.32	7.11	38809.52	87.78
			0.1	5992.42	3.3	19766.49	40.74
	10132	M=4	0.1	6597.16	4.52	29825.77	55.80
			0.1	6597.16	4.52	29825.77	55.80

The PDP of Type 2 is typically about $1.3\times$ that of Type 3, and the number of transistors is also $2.5\times$ times higher. The output data range of the balanced MAC is $2.2\times$ larger than that of the unbalanced MAC. In terms of overall performance, the balanced Type-3 has more advantages than an unbalanced Type-2. A balanced representation, however, is slower. It usually is twice longer delay, limiting its application to high-speed scenarios. In summary, the unbalanced MAC operates at a higher frequency but requires larger circuit area and power. Alternatively, the balanced circuit shows higher efficiency but the maximum frequency is lower.

III. APPROXIMATE TERNARY MULTIPLY-ACCUMULATOR

In this section, we describe ternary approximate computation, including ternary adders, ternary 4-2 compressors, 2-trit ternary multipliers, and middle-scale ($\sim 1,000$ transistors) ternary approximate multiply accumulator units. All of the ternary cells have been evaluated using 32 nm CNTFET technology.

A. 1-Trit Ternary Approximate Adder

Approximate multipliers can be commonly used for signal processing in edge applications where energy constraints are strict. Previous ternary multipliers utilized an unbalanced encoding scheme (0/1/2). For example, Tabrizchi et al. proposed approximate ternary multiplier using a 1-trit \times 1-trit approximate unbalanced ternary multiplier [28]. The circuit is based on approximate multipliers rather than adders, ensuring higher precision by employing fewer adders. This work focuses on balanced ternary scheme-based approximate circuits.

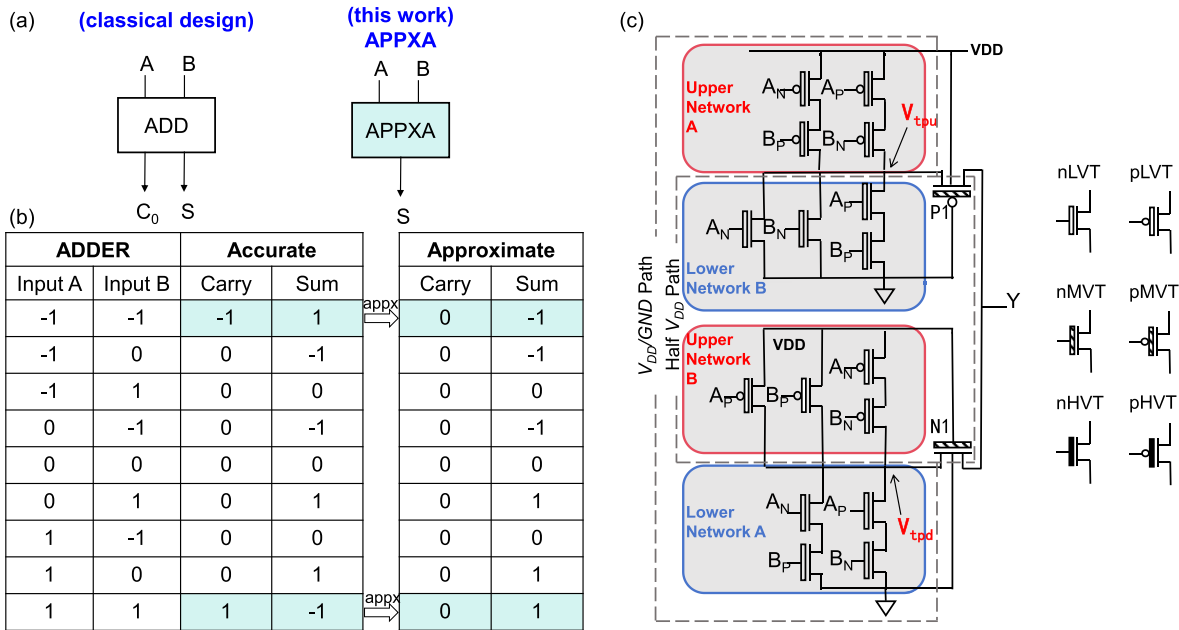


Fig. 2. (a) Circuit symbol of the classical adder and the proposed approximate adder. (b) Truth table of the ternary approximate adder. (c) Schematic and operation a mechanism of the approximate ternary adder.

Fig. 2(a) depicts a classical 1+1 balanced ternary accurate adder (2 inputs, 2 outputs) and the proposed approximate adder (2 inputs, 1 output). The truth table for the balanced approximate adder (APPXA) is shown in **Fig. 2(b)**. Two computation errors out of nine input combinations exist; namely, the approximation occurs in the case of “1+1” and “-1-1,” in which the circuit responsible for carry generation is eliminated. To analyze the significance of errors, two commonly used error metrics are employed, namely the Mean Absolute Percentage Error (MAPE) and the Mean Percentage Error (MPE).

$$\text{MAPE} = \frac{1}{n} \sum \left| \frac{\text{acc}_i - \text{appx}_i}{\text{acc}_i} \right|$$

$$\text{MPE} = \frac{1}{n} \sum \frac{\text{acc}_i - \text{appx}_i}{\text{acc}_i}$$

where acc_i represents the accurate result, appx_i denotes the approximate result, and n is the number of different input combinations. MAPE is the expected value of the errors encountered during the approximation process; thus, precision is highly correlated with MAPE. MPE provides information about whether the errors are biased. If the absolute value of the MPEs is equal to the MAPE, the approximation results are completely biased toward one side. The MPE of our proposed circuit is 11.1%, and the MPE is 0.

Fig. 2(c) depicts a gate-level schematic of the APPXA, employing V_{DD} , $V_{DD}/2$, and Gnd to represent respectively “1,” “0,” and “-1.” Both p-type and n-type CNTFETs are utilized with three different threshold voltages [20].

Using an improved Quine-McCluskey algorithm [29], we devised logic gates with the minimum number of transistors, resulting in higher energy efficiency. This structure consists of two paths, V_{DD}/Gnd path and half V_{DD} path, which contains two pass transistors (P1, N1). The V_{DD}/Gnd path

consists of upper/lower Network A, and the half V_{DD} path consists of upper/lower Network B [29].

The primary functionalities of APPXA are as follows: if inputs $[A, B] = [1, -1]$ or $[0, 0]$ or $[1, -1]$, the output Y is 0; if inputs $[A, B] = [1, 0]$ or $[1, 1]$ or $[1, 0]$, the output Y is 1; if inputs $[A, B] = [-1, 0]$ or $[-1, -1]$ or $[0, -1]$, the output Y is -1. The computation steps are:

Step 1: Signals A and B go through initial transformations via NTI and PTI (using A as an example): when the input A is -1, the outputs are $A_N = 1$ and $A_P = 1$; when the input A is 0, the outputs are $A_N = -1$ and $A_P = 1$; when the input A is 1, the outputs are $A_N = -1$ and $A_P = -1$.

Step 2: Process the signals A_N , A_P , B_N , and B_P through two stages to generate the intermediate V_{tpu} and V_{tpd} . For the upper network, if $A_N = 1$ or $B_N = 1$ or $A_P = B_P = 1$, the upper network outputs $V_{tpu} = -1$; if $A_N = B_P = -1$ or $A_P = B_N = -1$, the upper network outputs $V_{tpu} = 1$. For the lower network, if $A_P = -1$ or $B_P = -1$ or $A_N = B_N = -1$, the lower network outputs $V_{tpd} = 1$; if $A_N = B_P = 1$ or $A_P = B_N = 1$, the lower network outputs $V_{tpd} = -1$.

Step 3: V_{tpu} and V_{tpd} influence P1 and N1, generating the final output. If $[A, B] = [1, 0]$ or $[1, 1]$ or $[0, -1]$, $V_{tpu} = V_{tpd} = V_{DD}$, the gate voltage of transistor P1 is Gnd, P1 is ON, the gate voltage of transistor N1 is at V_{DD} , N1 is ON, therefore, the output $Y = \text{logic } 1$. If $[A, B] = [-1, 0]$ or $[-1, -1]$ or $[0, -1]$, $V_{tpu} = V_{tpd} = \text{Gnd}$, the gate voltage of transistor P1 is Gnd, P1 is ON; the gate voltage of transistor N1 is at V_{DD} , N1 is ON, therefore the final output is $Y = \text{logic } -1$. Note that in these two scenarios, there is no source-drain voltage drop across N1 and P1. With both N1 and P1 only marginally conducting, static power consumption is significantly reduced. If $[A, B] = [1, -1]$ or $[0, 0]$ or $[1, -1]$, $V_{tpu} = \text{Gnd}$, $V_{tpd} = V_{DD}$. The gate voltage of transistor P1 is at Gnd, P1 is ON; the gate voltage of transistor N1 is at V_{DD} , N1 is ON. The output is

TABLE IV
PERFORMANCE COMPARISON OF CLASSICAL
WORK AND THIS WORK (@ 32 NM CNT)

@ 1 GHz	classical design (ADD)	this work (APPXA)
Transistor Count	56	18
Avg power (μ W)	0.021	0.011
Worst Delay (ps)	174	66
PDP (aJ)	3.654	0.726
MAPE (%)	0	11.1
MPE (%)	0	0

the series voltage drop across P1 and N1, which is at $V_{DD}/2$, thus the final output $Y = \text{logic } 0$. Note that both P1 and N1 are also partially on.

Transient simulations using HSPICE and the 32 nm CNT-FET library are performed to validate the performance of the proposed circuit [20]. **Table IV** outlines the characteristics of the proposed circuit in terms of the transistor count, average power consumption, worst case propagation delay, PDP, MAPE, and MPE. As compared with the classical circuit, the proposed APPXA demonstrates superior efficiency with reductions in transistor count, average power consumption, and worst case delay. Specifically, the APPXA transistor count is 18, which is 67.9% less than the classical circuit. The propagation delay is significantly reduced by 62.1% to 0.066 ns, and the average power consumption is lowered by 47.6%, resulting in a 80.1% reduction in the total PDP.

B. Ternary 4-2 Compressor

The addition process of four inputs to two outputs is a 4-2 compression calculation. In balanced ternary logic, the three input adder circuit produces outputs ranging from $-3 = (-1, 0)_3$ to $3 = (1, 0)_3$, whereas a 2-trit ternary number spans from $-4 = (-1, -1)_3$ to $4 = (1, 1)_3$. Therefore, unlike binary logic, ternary logic can accommodate the addition of four 1-trit ternary numbers, resulting in only a 2-trit ternary result (4-2 compressor). We can optimize the 4-2 compressor through equivalence transformations by employing APPXA or replacing logic gate modules, such as CONS and ANY, with their inverted functions, NCONS and NANY, within the circuit structure. Similar to the binary ANY and OR gates, the inversion of these functions in ternary logic also require fewer transistor count and shortens the critical path. Consequently, this approach minimizes the number of gates and simplifies the overall circuit design, leading to reductions in both area and power consumption, while maintaining the required functionality for balanced ternary addition operations. Therefore, the 4-2 compressor is preferred for increasing data density and minimizing the transistor count in ternary adders [30].

Fig. 3(a) depicts circuit diagram of the standard 4-2 compressor, where Four 1-trit input signals are added to produce two 2-trit output signals by using THAs. The addition two 2-trit signals can be obtained by a 2-trit adder tree. The output

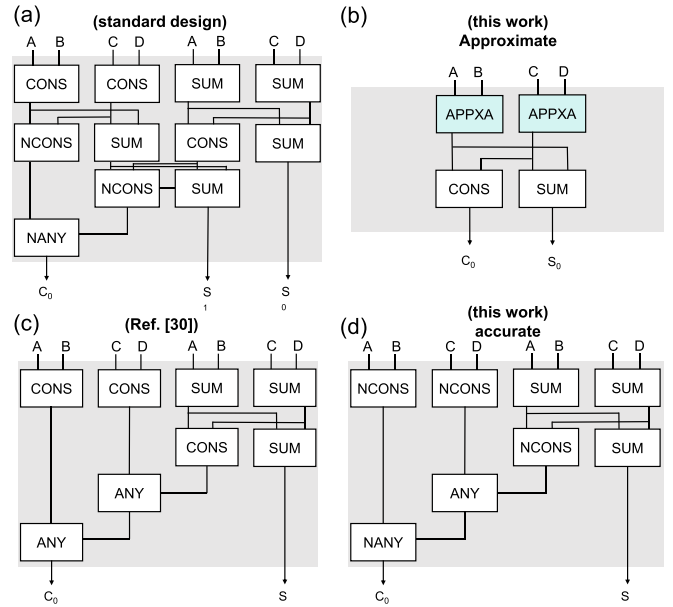


Fig. 3. Circuit diagram of the (a) standard 4-2 compressor, (b) original approximate 4-2 compressor, (c) 4-2 compressor process based on ref. [30], (d) proposed accurate 4-2 compressor. Note: The SUM, NCONS, and NANY are all components of the balanced ternary full adder [22].

signal is 3-trit. Since the maximum four 1-trit addition is 4 and the minimum is -4, the output can be represented within a 2-trit design.

Fig. 3(b) shows the proposed approximate 4-2 compressor, where the four 1-trit input signals are initially added using two approximate adders. Next, the two 1-trit output signals are passed from the previous level through the 1-trit addition process to obtain the final result (a 2-trit output signal), which is achieved using only one THA.

Regarding the accurate 4-2 compressor, a standard 4-2 compressor circuit is relative complex. To reduce the number of transistors, Yoon et al. proposed an optimized topology [30], as shown in **Fig. 3(c)**. We further optimize the 4-2 compressor circuit structure, as depicted in **Fig. 3(d)**. Note that the optimized 4-2 compressor differs by one ANY gate delay compared to the balanced TFA [22]. This improvement can significantly enhance the efficiency of an adder tree in terms of both delay and power.

The performance of the proposed circuit has been validated using HSPICE with 32 nm CNTFET library, and the simulated results have been shown in **Table V**. It can be seen that the proposed balanced approximate circuit achieves significant efficiency improvements over the standard 4-2 compressor: 68.5% fewer transistors, 42.4% lower power consumption, and 60.7% faster. These enhancements contribute to a notable decrease in the PDP, a 77.3% reduction. Additionally, the MAPE for the proposed ternary approximate scheme is 1.23%.

Table V compares the optimized accurate 4-2 compressors. Our circuit has 33.6% fewer transistors, 38.3% lower power, and 57.7% less delay, resulting in 74.0% lower total PDP. Additionally, as compared with the circuit proposed in [30], the number of transistors in our optimized 4-2 compressor is 194, marking a substantial decrease of 10.2%. Additionally,

TABLE V
PERFORMANCE COMPARISONS OF THE FOUR TYPES OF
4-2 COMPRESSORS (@32 nm CNT)

@ 1 GHz	classical design	Ref.[30]	(this work)	
			Accurate	APPX4-2
Transistor Count	292	216	194	92
Avg power (μ W)	0.752	0.610	0.464	0.433
Worst Delay (ns)	0.369	0.178	0.156	0.145
PDP (fJ)	0.277	0.109	0.072	0.063
MAPE (%)	0	0	0	1.23

the optimized 4-2 compressor achieves an 33.9% reduction in PDP.

Overall, the accurate 4-2 compressor offers a preferable option in those scenarios where precise computation is paramount. Alternatively, when a small decrease in accuracy is acceptable, utilizing the proposed approximate solution of a 4-2 compressor will increase overall circuit efficiency.

C. 1-Trit Ternary Approximate MAC Unit

The typical multiplication-accumulation operation requires both signed multiplication and signed accumulation. Therefore the balanced ternary scheme is more suitable for the signed MAC. A MAC unit contains two parameters: input vector lengths (M) and input data widths (N). Here we use the condition of M=4 and N=1 as an example to illustrate the results. The operational principle of MAC4_ACC_1trit (M=4, N=1) can be expressed as: $out = A_0 \times B_0 + C_0 \times D_0 + E_0 \times F_0 + G_0 \times H_0$. The operational process is divided into two steps. Four sets of partial products are obtained through the 1×1 multiplication using a balanced TMul. The four partial products are accumulated to obtain the final result (2-trit output signal) using a four input balanced ternary compressor (namely, the optimized 4-2 compressor described in the last section).

For the approximate multiplication-accumulation operations (M=4, N=1, MAC4_APPX_1trit), the operating principle can be expressed as: $out = appxa(A_0 \times B_0, C_0 \times D_0) + appxa(E_0 \times F_0, G_0 \times H_0)$, where appxa represents the approximate adder. The calculation process can also be divided into two steps. Firstly, four sets of partial products for the 1×1 multiplication are obtained using balanced TMul. Secondly, two APPXAs are used to accumulate the four partial products. A THA is cascaded to obtain the final result (2-trit).

The difference between approximate and accurate computing schemes is discussed in step 2. The approximate calculation uses approximate addition in the second step of the addition process. The MAC4_ACC_1trit unit directly employs a 4-2 compressor to generate a 2-trit output signal. Conversely, the MAC4_APPX_1trit unit utilizes two APPXAs to derive two 1-trit results, subsequently cascading a THA to yield the

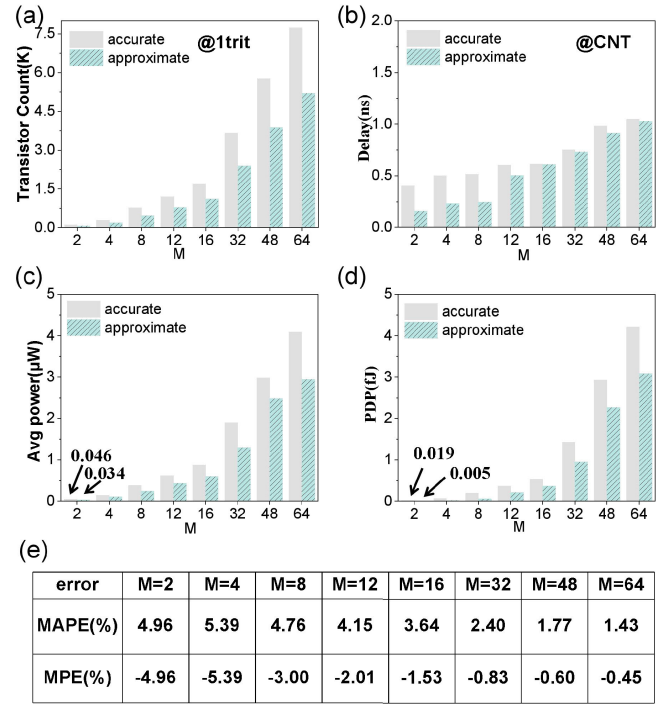


Fig. 4. Performance comparisons of (a) transistor count, (b) delay, (c) average power consumption, (d) PDP, and (e) error of MAPE and MPE at different vector lengths (M) in the 1-trit input MAC unit.

final 2-trit output signal. In the overall system, the accurate version requires 298 transistors, while the approximate version only requires 196 transistors.

The input vector length M represents the number of multiplication operations performed during each computational cycle. The multiplication-accumulation operations for the M=16 and N=1 (MAC16_1trit) calculation process is divided into two steps:

Step 1: Separating the sixteen 1-trit input signals into four MAC4_ACC_1trit units or MAC4_APPX_1trit units to produce four 2-trit temporary output signals.

Step 2: Using an adder tree, which consists of two 4-2 compressors, to sum the four 2-trit output signals and produce the final result (3-trit signal).

To assess the performance of our proposed method, we conducted transient simulations of the MAC circuits across various vector lengths (M=2, 4, 8, 12, 16, 32, 48, 64) in HSPICE using 32 nm CNTFET MOSFET library.

Fig. 4(a) describes a comparison of the transistor count of the approximate MAC and accurate MAC. The approximate version utilizes fewer transistors under different vector lengths of M. For example, when M=16, the transistor count of the accurate MAC unit is 1,692. Meanwhile, the transistor count of the approximate MAC unit is 1,123, a reduction of 33.6%.

Fig. 4(b) shows a comparison of the worst delay at different vector lengths. The delay of the approximate MAC is slightly lower than the accurate MAC. When M=16, the delay of the accurate MAC unit is 0.612 ns. The delay of the approximate MAC unit is 0.608 ns, a 0.7% reduction. Furthermore, the approximate MAC exhibits significantly lower average power consumption as compared with the accurate

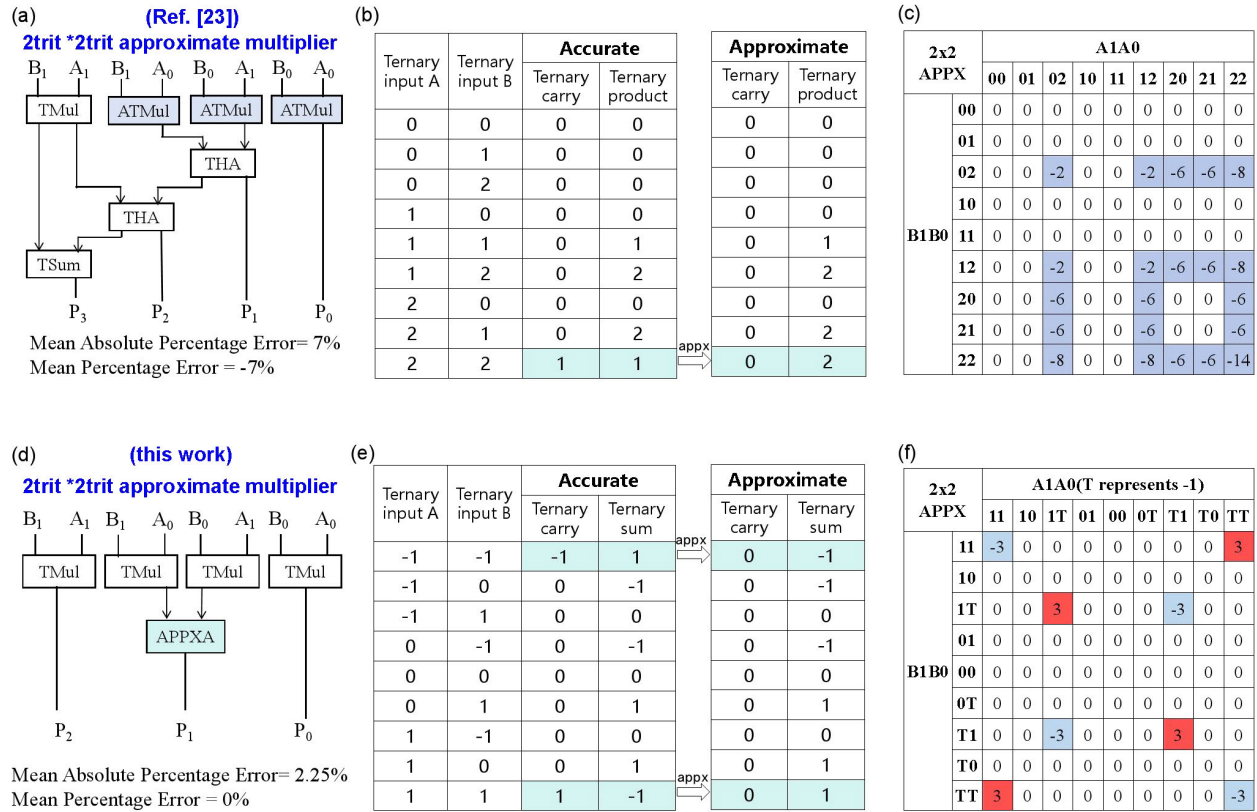


Fig. 5. (a) The 2-trit unbalanced approximate ternary multiplier proposed in [23]. (b) Error source: approximate computing in ternary multiplier. (c) Error distribution of the traditional 2-trit approximate multiplier. (d) The proposed 2-trit approximate ternary multiplier. (e) Error source: approximate computing in ternary adder. (f) Error distribution of the proposed 2-trit approximate multiplier. Note: ATMul is an approximate ternary multiplier, THA is a ternary half adder, and TSum is an unbalanced ternary summation. Error distribution = approximate value - accurate value.

MAC, as illustrated in Fig. 4(c). Specifically, when $M=16$, the accurate MAC consumes an average power of $0.88 \mu\text{W}$, whereas the approximate MAC consumes only $0.60 \mu\text{W}$, a 31.8% decrease. This result can be used to determine the MAC PDP based on the delay and average power consumption, as shown in Fig. 4(d). When $M=16$, the PDP of the accurate MAC unit is 0.539 fJ . The PDP of the approximate MAC is 0.366 fJ , 32.1% lower than the accurate MAC.

We conducted random number tests using Python to analyze the computational error. Fig. 4(e) displays the distribution of MAPE for the approximate MAC across 10^6 inputs and various M values. Observe that larger values of M produce lower MAPE errors. Specifically, at $M=16$, the MAPE error is 3.64%, while at $M=64$, the MAPE error drops to 1.43%.

D. 2-Trit Ternary Approximate Multiplier

A multi-trit multiplier serves as a fundamental building block for numerous complex logic circuits. Fig. 5(a) depicts the gate-level circuit diagram of a 2-trit \times 2-trit unbalanced ternary approximate multiplier [23], which maintains accuracy during the addition stage while applying approximation in the multiplication stage. The adder of THA and TSum is accurate, and the ATMul module is the core of the unbalanced ternary approximate multiplier. Fig. 5(b) presents the truth table used for the approximation method in the unbalanced approximate multiplier, or ATMul.

There is only one case for generating a carry signal: “ $2 \times 2 = 4$ (carry=1, product=1).” To simplify the calculation, the approximate strategy can be applied by “ $2 \times 2 = 2$ (carry=0, product=2).” Therefore, the 1-trit multiplication is carry-free. However, this strategy [23] introduces a negative computation bias in the multiplier, resulting in a relatively large system error, as shown in Fig. 5(c).

Fig. 5(d) shows the gate-level circuit of the proposed balanced ternary approximate multiplier, in which the accuracy is maintained in multiplication and the approximate is applied during the addition stage. As depicted in Fig. 5(a), the 2-trit \times 2-trit unbalanced ternary approximate multiplier consists of one 1-trit \times 1-trit unbalanced TMul, three 1-trit \times 1-trit ATMuls, two unbalanced THAs, and a TSum. The partial products generated by the multiplier are summed using unbalanced ternary adders. As shown in Fig. 5(d), the 2-trit \times 2-trit balanced ternary approximate multiplier consists of only one balanced ternary APPXA and four 1-trit \times 1-trit TMul modules.

Figs. 5(c) and 5(f) depict the error distribution of the 2-trit ternary approximate multipliers in the unbalanced ternary [23] and proposed balanced ternary schemes. For the balanced ternary approximate multiplier, errors only exist in 21 out of 81 input combinations, as shown Fig. 5(c). Reference [23] introduced the APPXA+9 compensation circuit to address this bias and enhance accuracy. Although the APPXA+9 circuit

TABLE VI
PERFORMANCE COMPARISON OF THREE
2TRIT MULTIPLIERS (@ 32 nm CNT)

@ 500MHz	2-trit \times 2-trit Approximate Multiplier		2-trit \times 2-trit Accurate Multiplier (classical design)
	(this work)	ref [23]	
Transistor Count	122	224	216
Avg power (μ W)	0.191	0.365	0.2767
Worst Delay (ns)	0.074	0.326	0.171
PDP (fJ)	0.012	0.119	0.047
MAPE (%)	2.25	7	0
MPE (%)	0	-7	0

improved accuracy to some extent, it still resulted in a MAPE of 3.4%.

In contrast, our proposed circuit not only achieves a more balanced error distribution, with 8 errors out of 81 input combinations and a lower MAPE of 2.25%, but also demonstrates greater efficiency in terms of area and power consumption, as shown in Fig. 5(f).

Table VI shows the simulation results of the proposed 2-trit \times 2-trit balanced and unbalanced approximate multiplier [23], and balanced accurate multiplier. As compared with the 2 \times 2 unbalanced approximate multiplier, the proposed circuit demonstrates a 45.5% reduction in transistor count, a 47.7% decrease in power consumption to 0.191 μ W, and a 77.3% reduction in delay to 0.074 ns, resulting in a 90% reduction in PDP. Additionally, our circuit also shows much smaller computational error. The total MAPE and MPE is respectively 2.25% and 0%, as compared to [23], where the two errors are respectively 7% and -7%. As compared with the 2-trit \times 2-trit balanced accurate multiplier, the proposed balanced solution also demonstrates a 43.5% reduction in transistor count, a 31% decrease in power consumption to 0.191 μ W, and a 56.7% reduction in delay to 0.074 ns, resulting in a 74.5% smaller PDP.

E. 2-Trit Ternary Approximate MAC Unit

To compare the performance of 2-trit balanced ternary accurate and 2-trit approximate computations, we evaluate an example ternary MAC with $M=4$ and $N=2$. The accurate multiplication-accumulation operation ($M=4$, $N=2$, MAC4_ACC_2trit) calculation process is divided into two steps:

Step 1: Eight 2-trit input signals are processed through four 2-trit \times 2-trit balanced accurate multipliers to yield four 4-trit output signals.

Step 2: An adder-tree with four 4-trit input signals is constructed from four of the optimized 4-2 compressors (4-inputs), three TFAs (3-inputs) and one THA (2-inputs). The adder-tree generates the 5-trit output signal.

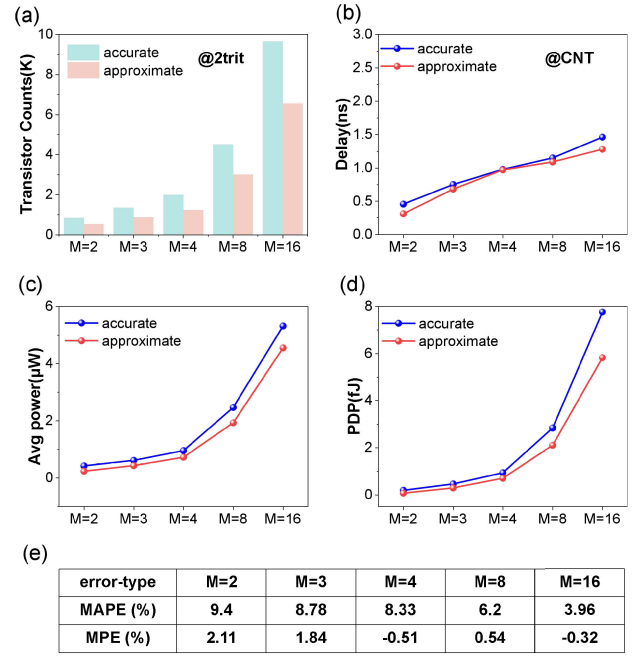


Fig. 6. Performance comparisons (@32 nm CNT) of (a) transistor count, (b) delay, (c) average power consumption, (d) power-delay product, and (e) statistical error of MAPE and MPE at different vector lengths (M) in the 2-trit input MAC unit.

Similarly, the approximate multiplication-accumulation operations ($M=4$, $N=2$, MAC4_APP_2trit) calculation process can be divided into two steps:

Step 1: Eight 2-trit input signals are processed through four 2-trit \times 2-trit balanced approximate multipliers to yield four 3-trit output signals.

Step 2: An adder-tree of four 3-trit inputs is constructed by using three of the optimized 4-2 compressors, and one THA. The adder-tree generates the 5-trit output signal.

To validate the performance of the 2-trit ternary MAC, transient simulations are conducted using HSPICE based on a CNTFET library [20]. Each test circuit is simulated with random inputs frequency = 1 GHz, supply voltage $V_{DD} = 0.9$ V, and temperature $T = 27$ °C. Fig. 6 presents the performance at different vector lengths ($M=2, 3, 4, 8, 16$). Fig. 6(a) shows that the transistor count of the approximate MAC is lower than the accurate MAC under different M. For example, when $M=4$, the transistor count of the accurate MAC is 1,994. The transistor count of the approximate MAC is 1,240, a 37.8% reduction.

Fig. 6(b) shows a delay comparison of the approximate MAC and accurate MAC at different M. The worst delay is a property of the circuit itself, mainly related to the circuit topology. The longer the critical path of the circuit, the higher the delay. The delay of the accurate MAC is slightly higher than the approximate MAC. When $M=4$, the delay of the approximate MAC is 0.97 ns. The delay of the accurate MAC is 0.978 ns, a 0.82% reduction.

Fig. 6(c) illustrates the average power consumption of the approximate MAC and accurate MAC at different M. The average power consumption of the approximate MAC is significantly lower than the accurate MAC. When $M=4$, the

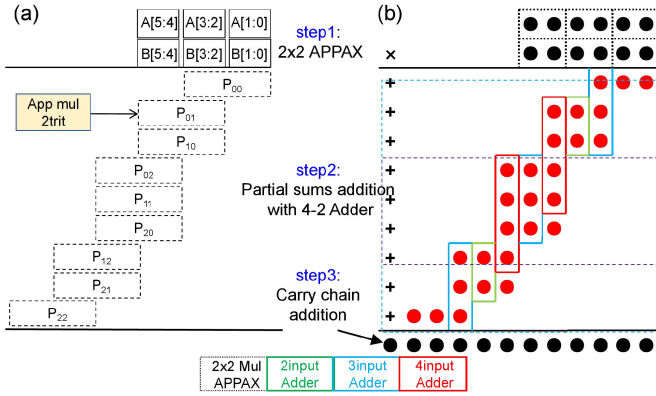


Fig. 7. (a) A 6×6 ternary approximation calculation multiplier circuit using a novel summation circuit. (b) Structure of 6×6 balanced approximate ternary multipliers with optimized Wallace-tree and 4-2 compressors.

average power consumption of the accurate MAC is $0.96 \mu\text{W}$. The average power consumption of the approximate MAC is $0.73 \mu\text{W}$, resulting in a 24% reduction. The PDP of the MAC based on the delay and average power consumption is shown in Fig. 6(d). When $M=4$, the PDP of the accurate MAC is 0.939 fJ . The PDP of the approximate MAC is 0.707 fJ , 24.7% less than the accurate MAC. The error statistical experiments use Python, and the MAPE/MPE results are shown in Fig. 6(e). Note that the MAPE error in the balanced approximate scheme is relatively low, particularly when the vector length of M is large.

IV. LARGE-SCALE APPROXIMATE TERNARY MULTIPLYING-ACCUMULATOR

In this section, we propose large-scale ($\sim 10,000$ transistor) ternary approximate logic circuits, including the 6-trit approximate multiplier and 6-trit multiply-accumulator unit.

A. 6-Trit Ternary Approximate Multiplier

The high trit-width ternary multiplier can be constructed from several low trit-width multipliers. As shown in Fig. 7(a), we cascade nine $2\text{-trit} \times 2\text{-trit}$ balanced ternary approximate multipliers, together with several ternary adders to accumulate the nine partial products (namely $P_{00}, P_{01}, \dots, P_{22}$), generating a $6\text{-trit} \times 6\text{-trit}$ balanced ternary approximate multiplier.

For the input signals of 6-trit A and B , each two adjacent trits are treated as one group. We produce $A[5:4]$, $A[3:2]$, $A[1:0]$, and $B[5:4]$, $B[3:2]$, $B[1:0]$. By using the 2×2 balanced ternary approximate multiplication as the basic unit, we obtain $P_{00} = A[1:0] \times B[1:0]$, $P_{01} = A[3:2] \times B[1:0]$, $P_{10} = A[1:0] \times B[3:2]$, and so on. The multi-group partial products are weighted sum using a ternary adder or 4-2 compressor, by which the computation error can be further adjusted once the approximate adder is included in the adder tree.

For the accumulation process, the Wallace-tree algorithm is utilized [22]. Fig. 7(b) illustrates the optimization scheme, where the dashed box denotes the 2×2 balanced approximate multiplier, the green box marks the two-input balanced THA, the blue box indicates the three-input TFA, and the red

TABLE VII
PERFORMANCE COMPARISON OF THREE 2TRIT MULTIPLIERS (@ 32 nm CNT)

@500MHz		error (%)	FET Number	Avg power (μW)	Delay (ns)	PDP (fJ)
ref. 23	Version A: accurate	0.000	6,532	8.117	2.518	20.44
	Version B: appx	9.176	4,876	6.605	2.346	15.50
	Version C: appx with bias	1.649	5,198	6.948	2.346	16.30
ours	Version A: accurate	0.000	3,946	14.07	1.250	17.58
	Version B: appx	3.540	3,284	8.222	1.055	8.67
	Version C: appx with 4-2 compressor	3.540	2,996	7.524	1.084	8.16

box designates the four-input adder, also known as the 4-2 compressor.

Step 1: Generate nine sets of partial products for the 6×6 multiplication using 2×2 approximate multipliers.

Step 2: Sum the partial products with two-input THAs (ternary half adders), three-input TFAs (ternary full adders), and four-input TCs (ternary compressors) until only three partial sums remain. Add the last three partial sums to obtain the final result using a carry-chain adder composed of two-input THAs, three-input TFAs, and four-input TCs.

Using the established sub-circuits of the 2×2 multiplier, partial products are initially generated with this multiplier, then compressed with a Wallace tree structure utilizing THAs and TFAs, and finally summed with a carry look-ahead adder to produce the final result. Notably, the 4-2 compressor reduces the number of compression stages in the Wallace tree and incorporates a 4-2 adder into the final addition carry chain, lowering the total number of adders from 23 to 15.

To validate the performance of the $6\text{-trit} \times 6\text{-trit}$ multiplier, transient simulations are conducted using HSPICE with the 32 nm CNTFET library [20]. Each test circuit is simulated with random inputs frequency = 0.5 GHz, supply voltage $V_{DD} = 0.9 \text{ V}$, and temperature $T = 27^\circ\text{C}$.

The simulation results are listed in Table VII, in which the MAPE error rate, transistor count, average power consumption, worst case propagation delay, and PDP are mentioned. As compared with the $6\text{-trit} \times 6\text{-trit}$ balanced ternary accurate multiplier, the PDP of the proposed circuit with APPX method is reduced by 50.6%. Compared to the 6×6 unbalanced ternary multiplier [23], our designs (versions A/B/C) achieve a 39.6% reduction in transistor count relative to the accurate version A, 32.6% relative to the approximate version B, and 42.4% relative to the optimized approximate version C. Furthermore, the proposed circuit demonstrates much less computational error with the MAPE error around 3.54% and MPE error = 0%. The efficiency of our circuit can be further improved by using approximate adders in the adder-tree.

TABLE VIII
PERFORMANCE COMPARISONS OF THE ACCURATE
AND APPROXIMATE MAC (@ 32 nm CNT)

@100MHz		Type	error (%)	FET Number	Delay (ns)	Avg Power (μ W)	PDP (fJ)
4 trit MAC	M=2	ACC	0.00	3,858	0.93	1.788	1.66
		APPX	1.91	3,138	0.89	1.501	1.34
	M=3	ACC	0.00	5,940	1.40	2.986	4.18
		APPX	1.65	4,860	1.34	2.548	3.41
	M=4	ACC	0.00	8,386	1.54	4.472	6.89
		APPX	1.02	6,946	1.49	3.939	5.87
6 trit MAC	M=2	ACC	0.00	9,090	1.23	4.89	6.01
		APPX	1.73	7,502	1.1	4.201	4.62
	M=3	ACC	0.00	13,794	1.58	8.046	12.7
		APPX	1.55	11,412	1.53	6.837	10.5
	M=4	ACC	0.00	19,154	1.81	11.62	21.0
		APPX	1.01	15,978	1.46	9.957	14.5

B. 6-Trit Ternary Approximate MAC Unit

To compare the performance of balanced ternary accurate computations with 6-trit approximate computations, we evaluate an example ternary MAC with $M = 3$ and $N = 6$. The accurate and approximate multiplication-accumulation operations ($M=3$, $N = 6$, MAC3_6trit) calculation process contains two steps. Step 1: Six 6-trit input signals are processed through three $6\text{-trit} \times 6\text{-trit}$ balanced accurate or approximate multipliers to produce three 12-trit output signals. Step 2: An adder tree composed of three 12-trit inputs is constructed from eleven 4-2 compressors and one TFA. The adder tree generates a 13-trit output signal.

Table VIII compares the performance of 4-trit and 6-trit accurate/approximate MACs at different vector lengths ($M=2, 3, 4$). Note that the number of transistors of the approximate MAC is lower than the accurate MAC under different M and same N . For example, when $N=6$ and $M=4$, the number of transistors of the accurate MAC is 19,154, while that of approximate is 15,978, corresponding to 16.6% reduction. The average power consumption of the approximate MAC is lower. When $N=6$ and $M=4$, the average power consumption of the accurate MAC is $11.62 \mu\text{W}$, while that of the approximate MAC is $9.957 \mu\text{W}$, corresponding to 14.3% reduction. Besides, the delay of the accurate MAC is 1.81 ns, while the delay of the approximate MAC is only 1.46 ns, a 19.3% reduction.

Finally, the PDP of the accurate MAC is 21.0 fJ. The PDP of the approximate MAC is 14.5 fJ, 30.95% lower. For the computational error, we carried out error statistical experiments using Python with 10^6 random inputs. The MAPE error of the approximate MAC is around 1.01%.

V. EVALUATION ON TERNARY NEURAL NETWORK

In this section, we demonstrate the application of ternary computing circuits (@32 nm CNT) on accelerating the ternary

neural network as an example. Meanwhile we compare the energy efficiency of ternary circuits with that of binary circuits when executing the same function.

A. Ternary Neural Network

A ternary neural network (TNN) is a type of artificial neural network where the weights and activations are represented using ternary values—typically +1, 0, and -1.

This is in contrast to traditional neural networks, which use floating-point numbers for their weights and activations. The advantages of ternary neural networks rely on two aspects. Firstly, ternary values can be stored more efficiently than floating-point numbers, leading to significant reductions in memory usage. Secondly, arithmetic operations on ternary values can be simpler and faster compared to those on floating-point numbers, potentially leading to faster computation times.

Ternary neural networks can be more energy-efficient. However, the current ternary neural networks are implemented using binary logic circuits, which leads to a decrease in efficiency.

Firstly, the inputs and outputs of ternary-valued neural networks are ternary, with only three states: -1, 0, and 1. Typically, this requires two wires to carry the signal. However, such 2-bit wires can carry up to four states, resulting in a decrease in state space utilization. If complex coding schemes are used to improve the utilization of input states, additional decoding overhead would be introduced. On the other hand, ternary circuits can perfectly solve this problem by achieving 100% utilization of input and output states.

Secondly, in the current implementation of ternary computing circuits, the utilization of information states within each module is low. As shown in **Fig. 8**, the input and output of S1 contains only 3 states (namely -1, 0 and 1). While 2 bit of wires are practically used, corresponding to 75% utilization efficiency in S0 and S1. As to S2, the total output of S2 will be five states, namely the minimum value is -2, and the maximum value is 2, requiring a 3-bit port to carry it. Since a 3-bit port can accommodate eight states, three of them remain unused, resulting in an actual utilization rate of 62.5%. In subsequent addition trees, a similar situation occurs. For instance, the output range of S3 is $[-4, 4]$, while the 4-bit output wire can accommodate sixteen states, 7 of them remain unused, resulting in an actual utilization rate of 56.25%.

In fact, using binary circuits to process ternary information inputs can never achieve 100% utilization efficiency, no matter what circuit connection topology is employed. This is because the range of data after summing N ternary inputs is symmetric, from $[-N, N]$. However, the range of two's complement representation in binary circuits is asymmetric, from $[-\text{Max}-1, \text{Max}]$. This mismatch results in wasted status bits in ternary computations within binary circuits of any bit width.

B. Performance Comparisons of Ternary and Binary Logic

To make a fair comparison between the efficiency of ternary and binary systems, both similarly sized ternary and binary adder trees are evaluated. All of the inputs are limited to $[1, -1, 0]$; thus, the final outputs range from -16 to 16.

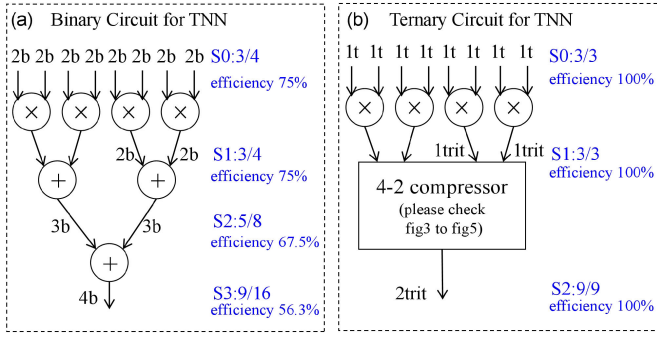


Fig. 8. Efficiency analysis of binary circuit and ternary circuit for the acceleration of Ternary Neural Network. The binary circuit can never achieve 100% utilization efficiency.

TABLE IX
PERFORMANCE COMPARISONS BETWEEN TERNARY
AND BINARY CIRCUITS (@ 32 nm CNT)

@32 nm CNT MAC16: inputs -1/0/1		delay (ns)	@50 MHz (μ W)	@200 MHz (μ W)	@500 MHz (μ W)	@1 GHz (μ W)
Ternary Circuit (Approximate)		0.543	0.349	1.022	1.961	3.575
Ternary Circuit (Accurate)		0.623	0.433	1.387	2.425	5.382
Binary circuit	V_{th} 0.687V	0.660	0.291	1.660	4.722	6.384
	V_{th} 0.618V	0.450	0.312	2.149	5.633	8.219
	V_{th} 0.506V	0.190	0.531	2.847	7.885	10.58
	V_{th} 0.428V	0.150	0.958	3.363	9.175	11.87
	V_{th} 0.371V	0.128	1.379	4.085	10.61	13.05
	V_{th} 0.323V	0.111	1.867	5.043	11.93	14.95

This task is commonly seen in the ternary neural networks. Ternary MAC16_ACC and ternary MAC16_APPX have been proposed in Section III. Binary MAC16_ACC is based on standard adder-tree. Considering that the performance of a binary circuit is dominated by the device threshold voltage, various threshold-voltage of the CNTFET have been evaluated as listed in Table IX.

As to the circuit delay, it can be seen that binary circuit shows higher frequency and lower delay in most cases, except the highest threshold voltage of $V_{th} = 0.687$ V. Such results suggest that the timing ternary computing is a drawback.

As to the power, ternary circuit is always better. Take the 200 MHz simulation result as an example, the power of the approximate and accurate ternary MAC unit is 1.022 μ W and 1.387 μ W, respectively. While the binary circuit shows relative larger power consumption ranging from 1.66 μ W to 5.04 μ W when the threshold voltage decrease from 0.687 V to 0.323 V.

Further calculations show that the power-delay product of binary logic circuits is about 10% to 20% lower than that of ternary circuits, mainly because binary circuits operate faster. However, if we consider the same operating frequency, which is a more conventional scenario, such as 1 GHz or 500 MHz or 200 MHz, ternary circuits have lower computational

power consumption. This highlights the advantages of ternary computing.

VI. SILICON BASED TERNARY LOGIC CIRCUIT

In this section, we verify the proposed circuit using Semiconductor Manufacturing International Corporation (SMIC) 180 nm Silicon process. The results show the ternary circuit demonstrate good CMOS-compatibility.

A. Silicon Based Ternary Logic Circuit

We have evaluated various ternary cells using CNTFET SPICE model. However, the CNTFET shows incompatibilities with current manufacturing processes, and is difficult to precisely control the diameter of the CNT to tune the threshold voltages. To demonstrate the manufacturability, compatibility, and reliability of ternary systems, an evaluation of commercially available silicon based ternary circuits is highly desirable (Supplementary Material S3). In our previous work [22], we successfully demonstrated the feasibility of realizing ternary logic gates utilizing three kinds of threshold voltages, low-threshold-voltage-N/PMOS (LVT_NMOS: $V_{th} = 0.292$ V; LVT_PMOS: $V_{th} = -0.1005$ V), middle-threshold-voltage-N/PMOS (MVT_NMOS: $V_{th} = 0.4185$ V; MVT_PMOS: $V_{th} = -0.424$ V), and high-threshold-voltage-N/PMOS (HVT_NMOS: $V_{th} = 0.756$ V; HVT_PMOS: $V_{th} = -0.695$ V).

We found that the operational frequency of a silicon ternary circuit is typically limited to ~ 200 MHz (or less). With the increase in module complexity, the system's frequency will gradually drop to approximately ~ 50 MHz. This low frequency is mainly because the PDK is not optimized for a ternary circuit, and it can be further improved in the future.

B. Performance Comparisons of Ternary and Binary Logic

Similar to that of Section-V, here we choose the ternary neural network as a test case. Fig. 9 depicts a comparison of the transistor count between different ternary and binary circuits. The transistor count of the 1-trit accurate MAC unit is 2,272, while the binary MAC is 4,212, corresponding to 46.06% fewer transistors. The power of the accurate ternary MAC unit is 64.83 μ W, while the high-threshold-voltage binary circuit is 69.62 μ W and the low-threshold-voltage binary circuit is 93.09 μ W. The energy efficiency of the approximate 1-trit circuit is 27.8% to 46% higher than the binary circuit. These results are listed in Table X.

Note that the disadvantage of ternary circuits is the lower speed. The operational frequency of a silicon based ternary circuit is usually limited to ~ 200 MHz or lower. Because many edge devices operate at a low frequency of 50 MHz, a ternary system remains effective. Furthermore, an accurate ternary circuit based on SMIC 180 nm process requires 45% less area and 30% less energy when compared with a binary circuit. Therefore, ternary logic can be an effective choice for low power edge computing

TABLE X
PERFORMANCE COMPARISONS BETWEEN
TERNARY AND BINARY CIRCUITS

SMIC-180 nm MAC16: inputs -1/0/1		Number of FET	Power (μ W)	Energy (fJ)
Silicon Ternary Circuit (this work)	LVT+MVT+ HVT device (Approximate)	1,614	50.26	1005.2
	LVT+MVT+ HVT device (Accurate)	2,272	64.83	1296.6
silicon binary circuit	LVT device (Accurate)	4,212	93.09	1861.8
	HVT device (Accurate)	4,212	69.62	1392.4

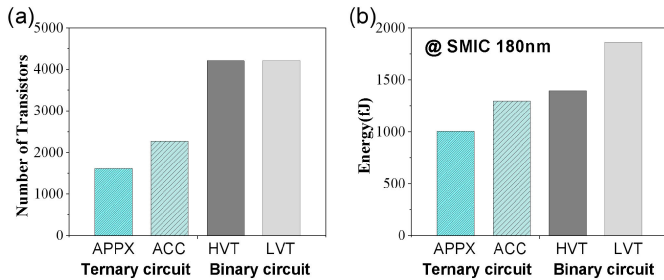


Fig. 9. Performance Comparisons of silicon based ternary and binary circuit on (a) Transistor counts and (b) Energy. Note that APPX and ACC mean the approximate and accurate, respectively. HVT and LVT represent respectively the high-threshold-voltage and low-threshold-voltage binary circuits.

VII. CONCLUSION

In this work, we focus on the ternary computing and developed various ternary arithmetic circuits including ternary approximate adders, 4-2 compressors, multipliers and multiply-accumulator unit. Compared with previous work, the proposed circuit exhibits smaller area, lower power-delay-product, and lower computational error rates, exhibiting superiority of the proposed method. Furthermore, we verify the design using standard silicon technology, and the simulation results show that the proposed ternary circuit is better than an equivalent binary circuit in terms of both area ($\sim 45\%$ less) and power ($\sim 30\%$ less), showing great potential for practical applications.

REFERENCES

- [1] J. Tang, T. Ma, and Q. Luo, "Trends prediction of big data: A case study based on fusion data," *Proc. Comput. Sci.*, vol. 174, pp. 181–190, Jun. 2020.
- [2] J. Wang, C. Xu, J. Zhang, and R. Zhong, "Big data analytics for intelligent manufacturing systems: A review," *J. Manuf. Syst.*, vol. 62, pp. 738–752, Jan. 2022.
- [3] T. N. Theis and H.-S. P. Wong, "The end of Moore's law: A new beginning for information technology," *Comput. Sci. Eng.*, vol. 19, no. 2, pp. 41–50, Mar. 2017.
- [4] M. H. Weik, "The ENIAC story," *Ordinance*, vol. 45, no. 244, pp. 571–575, 1961.
- [5] K. Donald, *The Art of Computer Programming*, vol. 2, 3rd ed., Reading, MA, USA: Addison-Wesley, 1997, ch. 4.
- [6] X. Wu and F. Prosser, "CMOS ternary logic circuits," *IEE Proc. G, Electron. Circuits Syst.*, vol. 137, pp. 21–27, Feb. 1990.
- [7] M. Khalid and J. Singh, "Memristor based unbalanced ternary logic gates," *Anal. Integr. Circuits Signal Process.*, vol. 87, no. 3, pp. 399–406, Jun. 2016.
- [8] M. Huang, S. Li, Z. Zhang, X. Xiong, X. Li, and Y. Wu, "Multifunctional high-performance van der Waals heterostructures," *Nature Nanotechnol.*, vol. 12, no. 12, pp. 1148–1154, Oct. 2017.
- [9] J. Shim et al., "Phosphorene/rhenium disulfide heterojunction-based negative differential resistance device for multi-valued logic," *Nature Commun.*, vol. 7, no. 1, p. 13413, Nov. 2016.
- [10] L. Lee et al., "ZnO composite nanolayer with mobility edge quantization for multi-value logic transistors," *Nature Commun.*, vol. 10, no. 1, p. 1998, Apr. 2019, doi: [10.1038/s41467-019-09998-x](https://doi.org/10.1038/s41467-019-09998-x).
- [11] G. Zhao et al., "Ternary logics based on 2D ferroelectric-incorporated 2D semiconductor field effect transistors," *Frontiers Mater.*, vol. 9, May 2022, Art. no. 872909.
- [12] W. Huang et al., "Ternary logic circuit based on negative capacitance field-effect transistors and its variation immunity," *IEEE Trans. Electron Devices*, vol. 68, no. 7, pp. 3678–3683, Jul. 2021.
- [13] R. A. Jaber, A. Kassem, A. M. El-Hajj, L. A. El-Nimri, and A. M. Haidar, "High-performance and energy-efficient CNFET-based designs for ternary logic circuits," *IEEE Access*, vol. 7, pp. 93871–93886, 2019.
- [14] J. M. Aljaam, R. A. Jaber, and S. A. Al-Máadeed, "Novel ternary adder and multiplier designs without using decoders or encoders," *IEEE Access*, vol. 9, pp. 56726–56735, 2021.
- [15] 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," *Comput. Electr. Eng.*, vol. 77, pp. 205–216, Jul. 2019.
- [16] A. Latha, S. Murugeswaran, and G. Yamuna, "Power optimized ternary arithmetic logic circuit using carbon nano tube field effect transistor," in *Proc. Int. Conf. Electron. Renew. Syst. (ICEARS)*, Mar. 2022, pp. 354–360.
- [17] Z. Zeng, G. Zhao, X. Wang, B. Kang Tay, and M. Huang, "Low power-delay-product ternary adder with optimized ternary cycling gates," in *Proc. 6th World Symp. Commun. Eng. (WSCE)*, Sep. 2023, pp. 98–102.
- [18] A. G. Asibelagh and R. F. Mirzaee, "Partial ternary full adder versus complete ternary full adder," in *Proc. Int. Conf. Electr. Commun., Comput. Eng. (ICECCE)*, Jun. 2020, pp. 1–6.
- [19] B. Srinivasu and K. Sridharan, "A synthesis methodology for ternary logic circuits in emerging device technologies," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 8, pp. 2146–2159, Aug. 2017.
- [20] C. Vudatha, A. Surya, S. Agrawal, and M. B. Srinivas, "Synthesis of ternary logic circuits using 2:1 multiplexers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 12, pp. 4313–4325, Dec. 2018.
- [21] B. S. Cherkauer and E. G. Friedman, "A hybrid radix-4/madix-8 low power signed multiplier architecture," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 8, pp. 656–659, Aug. 1997.
- [22] G. Zhao et al., "Efficient ternary logic circuits optimized by ternary arithmetic algorithms," *IEEE Trans. Emerg. Topics Comput.*, vol. 12, no. 3, pp. 826–839, Sep. 2024.
- [23] S. Kim, Y. Kang, S. Baek, Y. Choi, and S. Kang, "Low-power ternary multiplication using approximate computing," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 8, pp. 2947–2951, Aug. 2021.
- [24] S. Kim and Y. Kim, "High-performance and energy-efficient approximate multiplier for error-tolerant applications," in *Proc. Int. SoC Design Conf. (ISODC)*, Nov. 2017, pp. 278–279.
- [25] F. Zahoor, T. Z. A. Zulkifli, F. A. Khanday, and S. A. Zainol Murad, "Carbon nanotube and resistive random access memory based unbalanced ternary logic gates and basic arithmetic circuits," *IEEE Access*, vol. 8, pp. 104701–104717, 2020.
- [26] S. Kim, S. Lee, S. Park, K. R. Kim, and S. Kang, "A logic synthesis methodology for low-power ternary logic circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 3138–3151, Sep. 2020.
- [27] L.-M. Peng, Z. Zhang, and C. Qiu, "Carbon nanotube digital electronics," *Nature Electron.*, vol. 2, no. 11, pp. 499–505, Nov. 2019.
- [28] S. Tabrizchi, A. Panahi, F. Sharifi, H. Mahmoodi, and A.-H. A. Badawy, "Energy-efficient ternary multipliers using CNT transistors," *Electronics*, vol. 9, no. 4, p. 643, Apr. 2020.
- [29] S.-Y. Lee, S. Kim, and S. Kang, "Ternary logic synthesis with modified quine-McCluskey algorithm," in *Proc. IEEE 49th Int. Symp. Multiple-Valued Log. (ISMVL)*, May 2019, pp. 158–163.
- [30] J. Yoon, S. Baek, S. Kim, and S. Kang, "Optimizing ternary multiplier design with fast ternary adder," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 2, pp. 766–770, Feb. 2023.



Wanting Wen received the B.S. degree in information and computational science from Wuhan Institute of Technology, Wuhan, China, in 2022, where she is currently pursuing the M.S. degree. Her current research interests include VLSI physical design optimization.



Guangchao Zhao received the B.S. degree in micro-electronic science and engineering from Wuhan University, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include multi-value-logic and circuit design.



Wanbo Hu received the B.S. degree in integrated circuit design and integrated system from the Huazhong University of Science and Technology, China, in 2018, and the M.S. degree from the Southern University of Science and Technology, Shenzhen, China, in 2024. His research interests include memristor circuits and in memory computing circuit design.



Ziye Li received the B.S. degree in physics in material from Hebei University of Technology, China, in 2018, and the M.S. degree from the Southern University of Science and Technology, Shenzhen, China, in 2024. His research interests include memristor circuits and in memory computing circuit design.



Xingli Wang received the B.S. degree from Jilin University, China, in 2010, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2016. He is currently a Senior Research Fellow with CNRS International—NTU—Thales Research Alliance (CINTRA). He is working on the synthesis 2D materials and their heterostructures, such as MoS₂ and WS₂, and exploring their application in field effect transistors, tunneling devices, and thermoelectric devices. He has published 46 peer-reviewed articles. His current H-index is 25.



Eby G. Friedman (Life Fellow, IEEE) received the B.S. degree in electrical engineering from the Lafayette College, Easton, PA, USA, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Irvine, Irvine, CA, USA, in 1981 and 1989, respectively. He is the author of more than 500 articles, book chapters, and 19 patents, and the author or an editor of 18 books in the fields of high-speed and low-power CMOS design techniques, 3-D design methodologies, high-speed interconnect, and the theory and application of synchronous clock and power distribution networks. His current research and teaching interests include high performance synchronous digital and mixed-signal microelectronic design and analysis with application to high-speed portable processors, low-power wireless communications, and server farms.



Beng Kang Tay (Senior Member, IEEE) received the B.Eng. (Hons.) and M.Sc. degrees from the National University of Singapore in 1985 and 1989, respectively, and the Ph.D. degree from the School of Electrical and Electronic Engineering (EEE), Nanyang Technological University (NTU), Singapore, in 1999. He is currently a Full Professor with NTU. He is also the Associate Chair of the School of Electrical and Electronic Engineering, NTU, and the Deputy Director of the CNRS International—NTU—Thales Research Alliance (CINTRA). To date, he has published more than 400 journal articles with Google Scholar H-index of 60. His research interests include the synthesis and applications of low dimensional materials, such as carbon nanofilms, carbon nanotubes, and 2D materials (especially transition metal chalcogenides).



Shaolin Ke received the B.S. and Ph.D. degrees in physics from the Huazhong University of Science and Technology, Wuhan, China, in 2013 and 2018, respectively. He is currently an Associated Professor with Wuhan Institute of Technology, Wuhan. His research interests include topological photonics, nanophotonics, and VLSI physical design optimization.



Mingqiang Huang received the B.Eng. and Ph.D. degrees in physics from the Huazhong University of Science and Technology, Wuhan, China, in 2013 and 2018, respectively. From 2018 to 2019, he was a Research Fellow with Nanyang Technological University, Singapore, focusing on energy-efficient micro-electronics and logic circuits. Since November 2019, he has been with Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, as a Research Associate Professor. His current research interests include memristor and memristor circuits and artificial intelligence (AI) hardware accelerators.