JTE HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

XNOR-Popcount, an Alternative Solution to the Accumulation Multiplication Method for Approximate Computations, to Improve Latency and Power Efficiency

Van-Khoa Pham[1]*, Lai Le[2], Thanh-Kieu Tran Thi[1]
[1]*Ho Chi Minh City University of Technology and Education, Vietnam*
[2]*Renesas Design Vietnam*

*Corresponding author. *Email: khoapv@hcmute.edu.vn*

ARTICLE INFO

Received: 10/03/2024
Revised: 15/04/2024
Accepted: 15/04/2024
Published: 28/02/2025

KEYWORDS

Multiply–accumulate operation;
XNOR-popcount;
Adder;
Latency;
Power consumption.

ABSTRACT

Convolutional operations on neural networks are computationally intensive tasks that require significant processing time due to their reliance on calculations from multiplication circuits. In binarized neural networks, XNOR-popcount is a hardware solution designed to replace the conventional multiplied accumulator (MAC) method, which uses complex multipliers. XNOR-popcount helps optimize design area, reduce power consumption, and increase processing speed. This study implements and evaluates the performance of the XNOR-popcount design at the transistor-level on the Cadence circuit design software using 90nm CMOS technology. Based on the simulation results, for the same computational function, if MAC operation uses XNOR-popcount, the power consumption, processing time, and design complexity can be maximally reduced by up to 69%, 50%, and 48% respectively when compared to the method using conventional multipliers. Thus, the XNOR-popcount design is a useful method to apply to edge-computing platforms with minimalist hardware design, small memory space, and limited power supply.

Doi: https://doi.org/10.54644/jte.2025.1537

1. Introduction

In convolutional neural networks, the convolution operation with Multiply–Accumulate (MAC) requires complex computational hardware and high power consumption [1], [2]. The image pixels of the receptive field are multiplied with the kernel (training weights). This multiplication is repeated until the last pixel in the image, shifted by one pixel at a time [3]. As illustrated in Figure 1, assumed that the convolution operation processes an input image of size 19×19 with the receptive field and the kernel size of 4×4. To obtain a 16×16 feature map, 4096 times of multiplication, addition, and memory access are needed. If floating-point numbers represent each value in the image pixels of the receptive field, the convolution processing consumes a large amount of time and power due to the computation of multiplication on floating-point data and frequent data movement between memory and processor [4], [5]. Thus, if the data movement between memory and processor can be limited and the multiplication with complex hardware is replaced by an approximate calculation method, the computational processing performance will significantly increase [1], [6]. The Binary Neural Network (BNN) model uses binary values to represent training weights and input values to reduce the network model size while still achieving acceptable accuracy [7]-[9]. This helps save memory space and energy, and makes the model easily deployable on edge-computing platforms with limited power and hardware resources. This study analyzes the operation of the convolution on the BNN using the conventional multiplication and an approximate computation method. The operation of the two designs will be executed and analyzed using the 90nm CMOS microchip technology.

JTE
HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

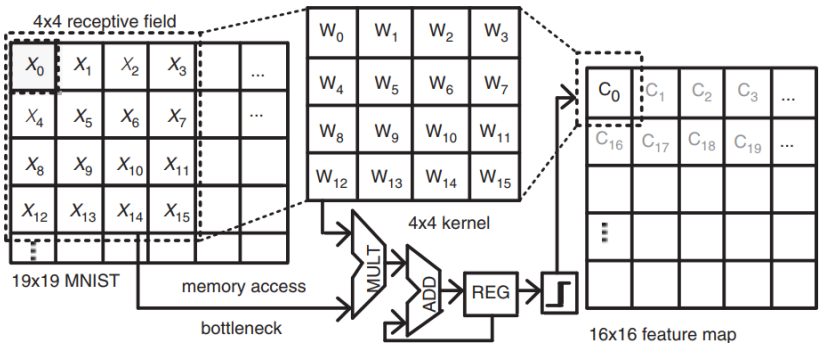

Figure 1. *The convolution multiplication operation on the input image of size 19×19 with the receptive field and the kernel size of 4×4*

2. Multiply–Accumulate (MAC) in Convolution Operation

Binarized Neural Networks (BNN) are a special case of Quantum Neural Networks (QNN) [7]-[9] where both the training parameters and activation signals are quantized into binary values, as illustrated in Figure 2. Thus, during the network training process, the algorithm changes the values of the parameters to become -1 or +1. The activation function used in the BNN is Sign(x), instead of using complex functions that are difficult to implement with hardware such as Sigmoid or ReLU. The Sign(x) function is used to determine the sign for the result of the Multiply–Accumulate operation (x) to satisfy the following equation:

$$f(x) = \text{sign}(Y) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (1)$$

Where:

$$Y = \sum_0^n (Xn * Wn) = X_0 * W_{00} + X_1 * W_{10} + X_2 * W_{20} + ... + Xn * W_{n0}$$

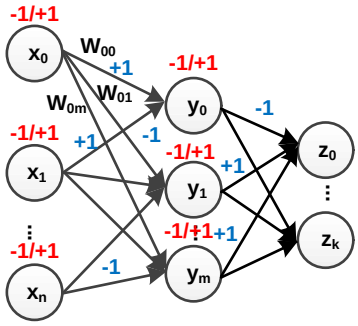

Figure 2. *Binarized Neural Networks*

By binarizing the training parameters, inputs, and output values of the activation function, the convolution operation, which includes multiplication and accumulation, requires simpler hardware than calculations on floating-point values. Multiplication and addition are the main components in the convolution operation. The multiply-accumulate unit consists of two types: sequential multiplication and parallel multiplication as shown in Figure 3. Considering the design in Figure 3a, the values of the input data X and the training parameters W are entered in sequential order, leading to a large delay. In this design, the number of addition operations will only depend on two factors: the size of the adder and the size of the accumulator, because as more additions are performed, the accumulated value is likely to increase and at this point, the value returned to enter the adder also increases. In the parallel multiplier in Figure 3b, the operations are performed almost simultaneously, so the calculation speed is significantly improved. However, the limitation is that the number of operations is only limited by the number of multipliers, to perform more operations, more multipliers and adders will have to be added, and this makes the design have a large size. Thus, the sequential multiply-accumulate will perform many

operations, however, the speed will be slow and the size is smaller, suitable for devices that need to calculate large data blocks and do not need high calculation speed. The parallel multiplier then improves the calculation time. However, it will be more limited in the number of operations, suitable for problems with a fixed number of operations and requiring high-speed response.

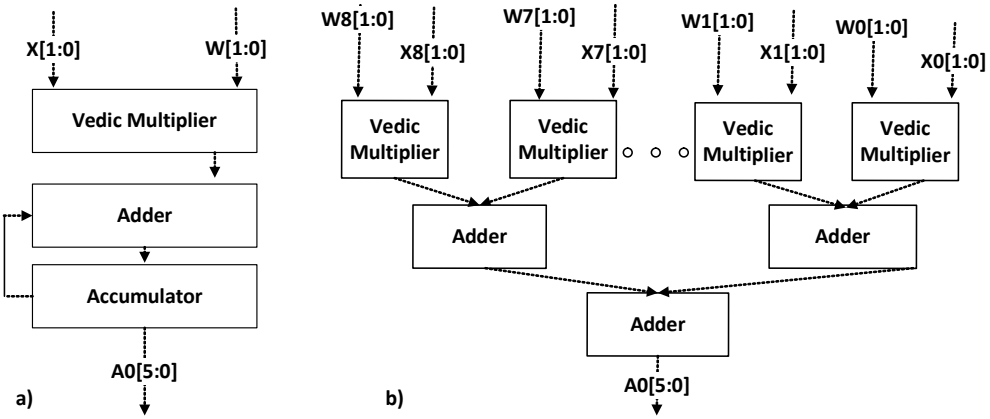

Figure 3. *Design of the multiply-accumulate operation a) sequential structure b) parallel structure*

In addition to arithmetic operations, the computer processor also contains bitwise operations. Compared to the hardware design of arithmetic operations, the hardware to execute bitwise operations is simpler [4]. Therefore, the power consumption and calculation time on bit processing operations will yield better performance [4], [6]. As illustrated in Figure 4, in the case of processing convolution multiplication with binary input data and training parameters, XNOR-popcount is an effective solution when achieving similar calculation results with low hardware cost.

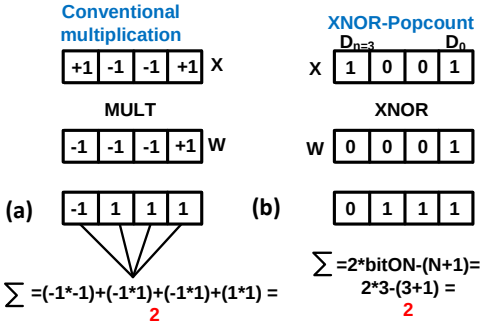

Figure 4. *The execution method and results of multiply-accumulate operations on binary numbers a) the MAC method b) the XNOR-popcount method*

The processing method of the XNOR-popcount design is represented by the following equation:

$$f(x) = 2p - N = 2\sum_{i=0}^{N-1} XNOR(X_i, W_i) - N \quad (2)$$

Here, X is the vector of the input image or the output of the activation function, W is the vector formed from the training parameters, and N is the vector length. Suppose the input value X[3:0] = {1; -1; -1; 1} and the training parameter W[3:0] = {-1; -1; -1; 1} as shown in Figure 4a, four multiplications and a 2-bit signed addition are used to obtain a convolution result of value 2. However, the memory space to store the training parameters and the hardware to process the convolution can be maximally simplified by using only 1 bit if an encoding operation is performed to remove the sign bit, converting the values +1 and -1 into 1 and 0 respectively as shown in Figure 4b. The general XNOR-popcount architecture design is illustrated in Figure 5a with three main operations: (1) performing XNOR processing on two 1-bit binary numbers, (2) performing Popcount to count the total number of 1 bits in the XNOR result, and (3) performing 2×S – N where S is the total number of 1 bits, and N is the fixed length of the vector. As illustrated in Figure 5b, the hardware needed to process XNOR-popcount

includes XNOR gates with two 4-bit inputs, an adder circuit to sum the one bits of the XNOR result, a left shift circuit to perform multiplication by two, and a 4-bit subtraction circuit. Especially to maximize hardware simplification, the second operand in the subtraction operation can be fixed by the value N that is the size of the filter and has been predefined. Compared to the multiply-accumulate method, the XNOR-popcount hardware in Figure 5b is simplified but the calculation result is equivalent. Assuming that W and X are 9-bit vectors. The binary adder performs the addition operation by performing binary addition on each corresponding pair of bits. To execute the accumulation operations, many adders must be used, the adders used can be up to dozens (if X or W is 18 bits) and many types of adders have different numbers of bits for each input such as 2 bits, 3 bits, 4 bits…

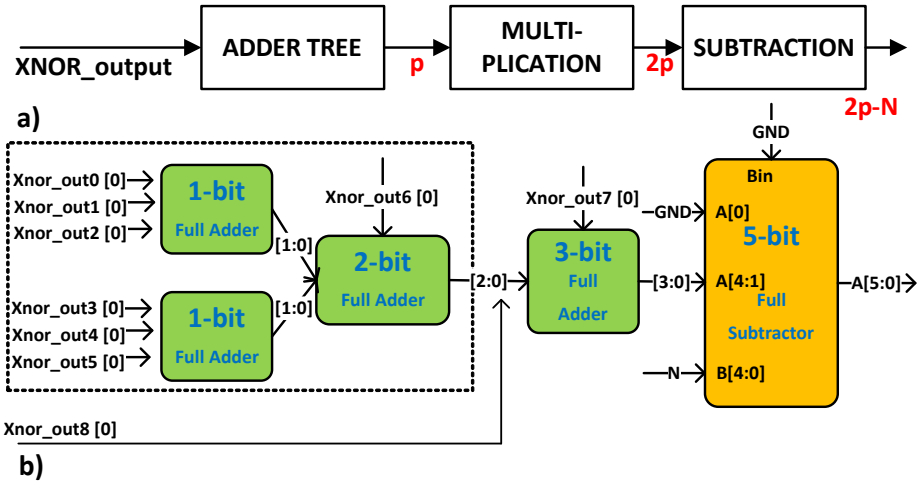

Figure 5. *The XNOR-popcount method a) Block diagram b) Circuit design*

3. Approximate Computation using XNOR-Popcount

As analyzed above, XNOR-popcount is designed from two main components: the XNOR block formed from XNOR gates and the accumulation block formed by an adder tree. Thus, optimizing the XNOR-popcount design will focus on these two main components. The full adder of two 1-bit numbers is an important component in the accumulation operation. Therefore, choosing an appropriate full adder design will greatly affect the operation and computational efficiency of the XNOR-popcounts design. In CMOS technology, choosing a suitable configuration, the power consumption of the design, and processing speed are very important aspects for the circuit to operate at high performance. Since XNOR-popcount is designed using many adders, optimizing the adder design can significantly reduce processing time, power consumption, and design size. This study surveys the design of some 1-bit Full-Adder in different configurations such as 54 transistors (54T) [10], 28 transistors (28T) [11], 10 transistors (10T) [12], eight transistors (8T) [12]. The 1-bit Full-Adder using 54T uses two 1-bit Half-Adder implemented by NAND logic gates. The adder using 54T is an adder that is quite commonly used in old designs due to the stability of the design; however, this adder has many disadvantages in terms of processing speed and size. The Full Adder 28T, 10T, and 8T are other configurations of the 1-bit Full-Adder to reduce design size, optimize delay, and power consumption. This study simulates the operation of the adders using Cadence Virtuoso software and 90nm CMOS technology at an operating frequency of 500MHz, operating voltage of 1V, and a room temperature environment of 27^{o}C.

Figure 6 shows the operation waveform of the adder designs. In which, the Sum output of the adder designs with 28 transistors, 10 transistors, and eight transistors are represented by S28T, S10T, and S8T respectively. Similarly, the Carry-out output of the adder designs will be C28T, C10T, and C8T respectively. Based on the waveform, it can be seen that all adder designs provide the proper logic level output. However, in different designs, the delay response of the output for the same input has a large difference. In the 8T adder, the logic level of the output is not stable at high frequencies. A solution to stabilize the logic level at the output of the 8T design is to add more buffers. However, this will double the design area. Because at least four transistors are needed for each buffer. In the case of the

conventional adder with 54 transistors with the corresponding output is S2HA and C2HA as well as the 28T and 10T adder, the Sum and Cout signals are stable. To evaluate the performance of the adder designs, this study conducts an evaluation of delay time and power consumption at an operating frequency of 500 MHz, operating voltage of 1V, and at a room temperature environment of 27ºC.

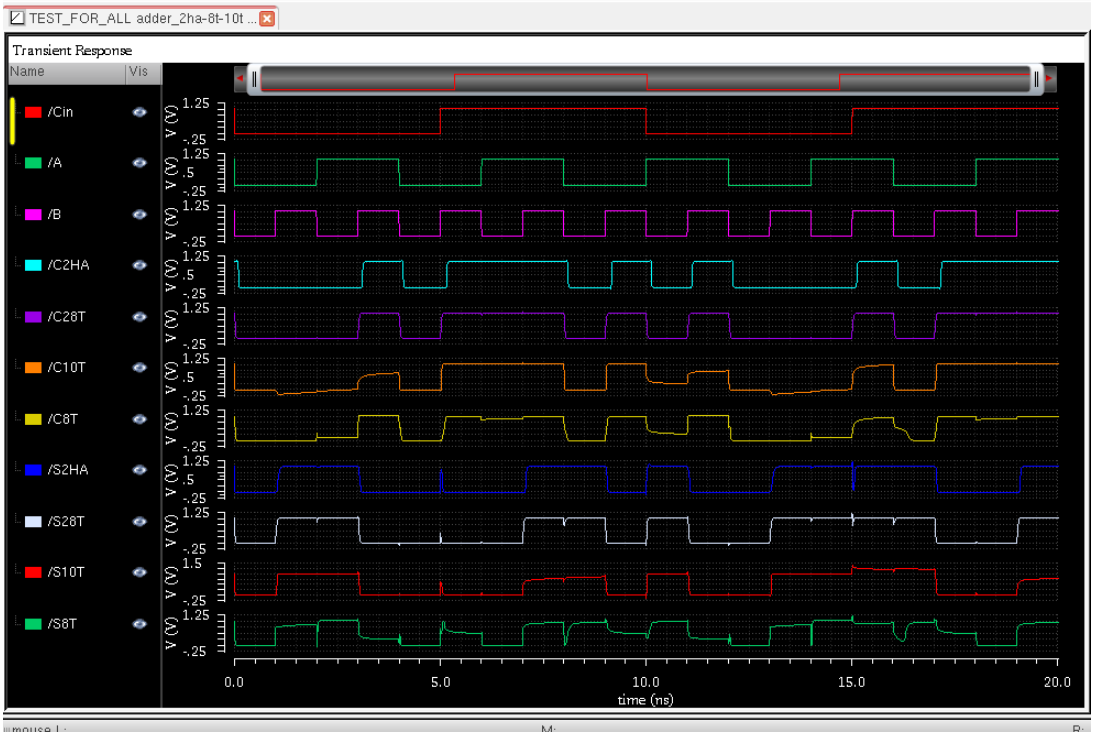

Figure 6. *The operation waveform of various 1-bit Full-Adder designs*

The simulation results on delay and power consumption are detailed in Figure 7. It can be seen that the adder with the design using 28T and 10T has a low delay at 25pS and 16pS respectively. Thus, the 10T adder has the lowest delay. The delay value of the 10T adder is eight times smaller than the traditional adder using 54T, and five times smaller than the adder using 28T. When compared to the 10T adder, the 28T adder has a low delay but requires a larger design area. Considering power consumption, the design of the 10T and 28T adders consumes 3µW and 6.9µW respectively at an operating frequency of 500 MHz. The power consumption of the 10T adder is only 8 times smaller than the conventional adder design with 54T consuming 24.1µW and less than twice as small as the 8T adder design. Thus, based on the analysis results of power consumption and delay time of the adder designs, the adder design with 10 transistors is suitable and chosen to implement the hardware of the XNOR-popcount design as it achieves the lowest power consumption and processing time.

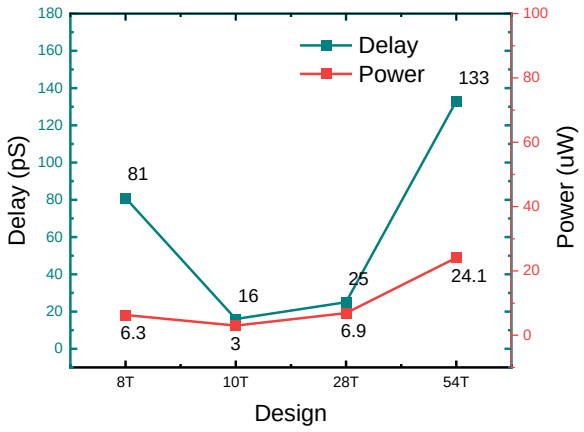

Figure 7. *Delay and power consumption of various 1-bit Full-Adder designs*

JTE
HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

The XNOR gate is a component that processes input values, and every input value must pass through the XNOR gate, so this design must be error-free and optimal. The conventional XNOR is built with 18 transistors (18T) with four NAND gates and one NOT gate. This design requires a large area, low processing speed, and high power consumption. This study investigates the configurations of the XNOR gate such as 14 transistors (14T), 12T, 10T, and 8T [13] to find the optimal configuration in terms of power consumption and delay time. This study simulates the operation of the XNOR designs using Cadence Virtuoso software and 90nm CMOS technology at an operating frequency of 500MHz, operating voltage of 1V, and a room temperature environment of 27°C. The waveform of the XNOR configurations is described in Figure 8. It can be seen that the waveform of XNOR 10T has a clearer logic level compared to the other configurations.

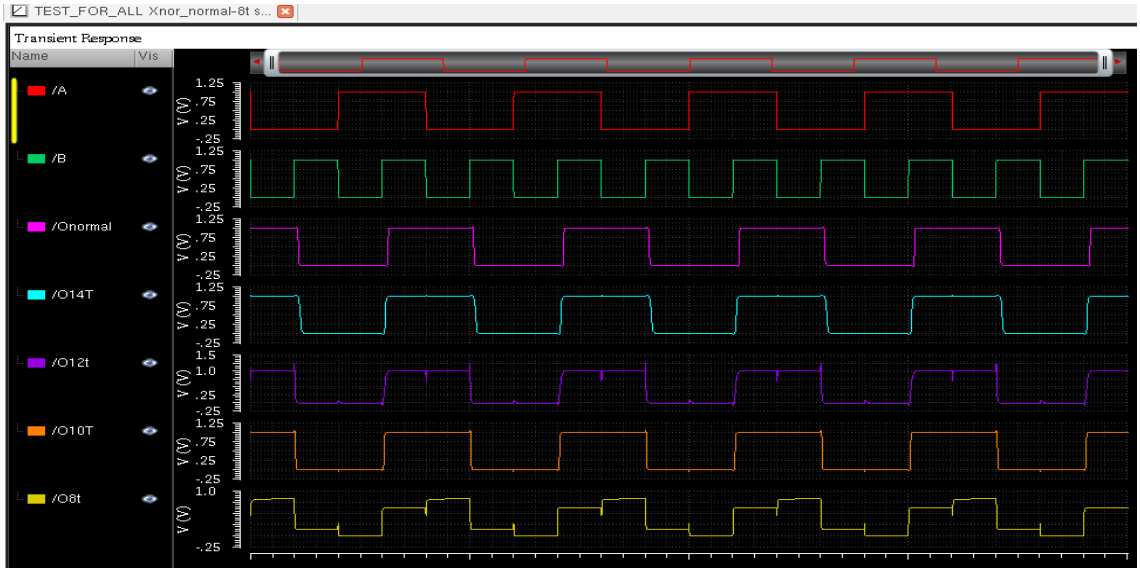

Figure 8. *The operation waveform of various XNOR gate designs*

Figure 9 shows the delay and power consumption parameters of various XNOR designs. The 12T, 10T, and 8T configurations have the lowest delay, about 7 times smaller than the conventional XNOR. The delay compared to the conventional XNOR varies by about 4 times with the 12T design, 3 times with the 10T design, and about 15 times with the 8T XNOR. The 8T XNOR design is small, only about 50% of the size of the conventional design, but with high power consumption. The 10T configuration satisfies the factors of design area, power consumption, and delay time.

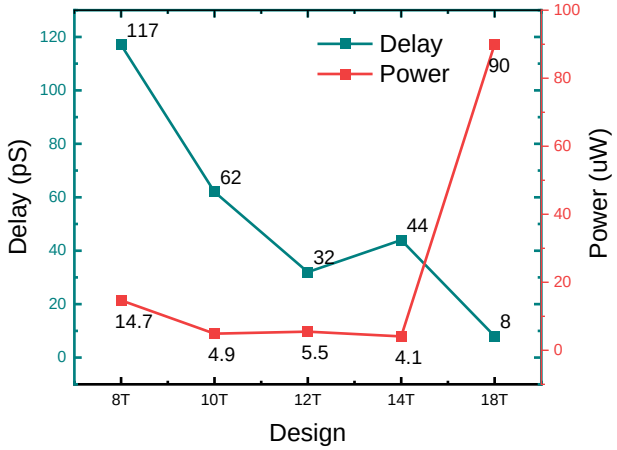

Figure 9. *Delay and power consumption of various XNOR gate designs*

3. Results

Based on the above analyses, this study uses the 10T configuration for both the 1-bit Full-Adder and the XNOR gate to construct a complete XNOR-popcount unit. To verify the correctness of the MAC and XNOR-popcount designs, this study used Cadence Virtuoso software and 90nm CMOS technology, under conditions of a 500MHz operating frequency, 1V operating voltage, and an ambient temperature of 27^{o}C to analyze the relationship between the inputs/outputs of the designs. For the same input signal, after passing through two blocks consisting of MAC and XNOR-popcount, it is observed that the outputs of the MAC and XNOR-popcount designs are equivalent. As shown in Figure 10, suppose at time t = 29.3ns, the value of the input data X[8:0] is {0,0,0,0,0,0,0,0,1} and the value of the training weight is W[8:0]= {0,0,0,0,0,0,0,0,0}, the output of the XNOR gate is O[8:0] = "111111110". Finally, the 4-bit register S [3:0] contains the XNOR-popcount result. In the above case, the register {Cout, S4: S0} = {0, 0, 0, 1, 1, 1}, which is the decimal value '7', corresponds to the result obtained from the multiply-accumulate method when the two inputs are X[8:0] = {-1; -1; -1; -1; -1; -1;-1; -1; +1} and W[8:0] = {-1; -1; -1; -1; -1; -1; -1; -1; -1}. In another case, when the inputs X[8:0] and W[8:0] are both {0,0,0,0,1,1,0,0,0}, the register {Cout, S4: S0} = {0, 0, 1, 0, 0, 1}, which is the decimal value '9'.

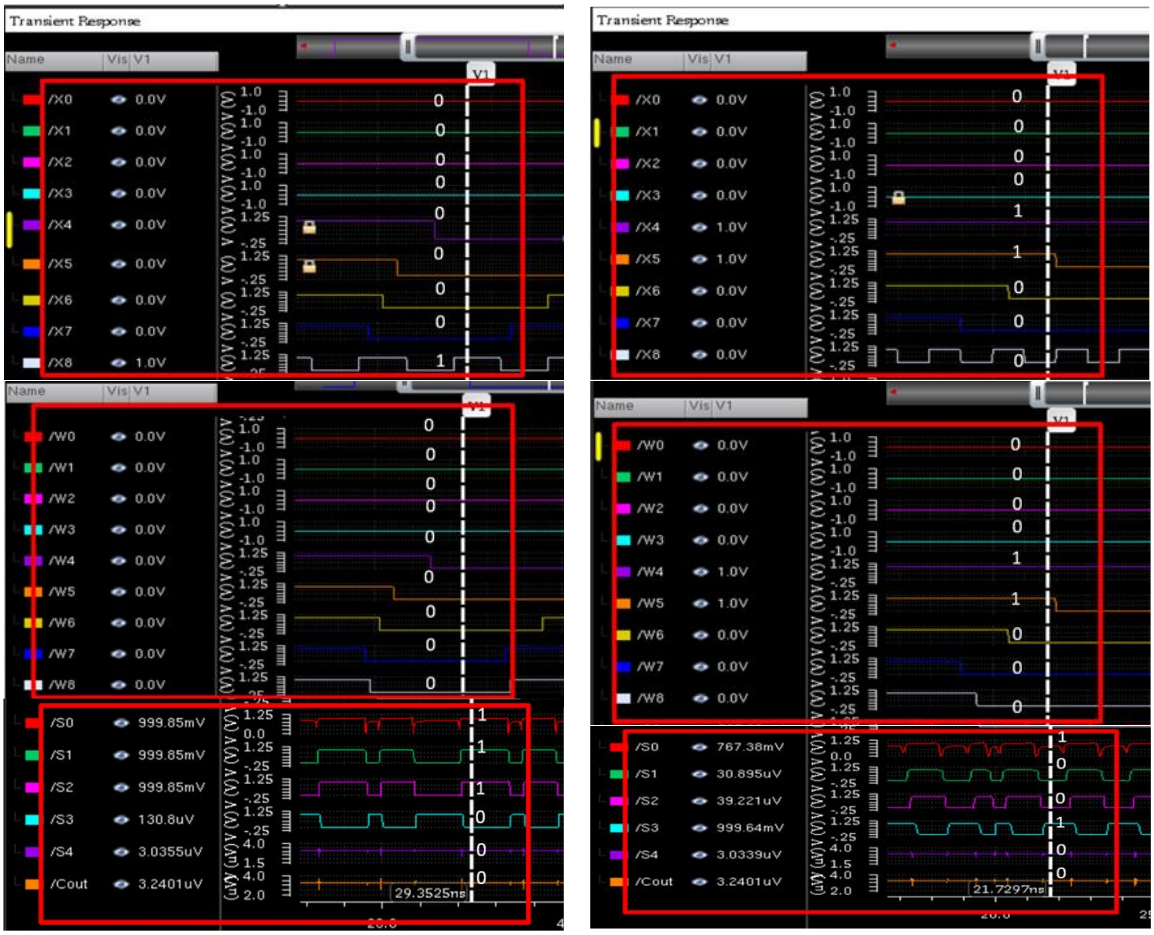

Figure 10. *The operation waveform of the XNOR-popcount implemented in this study*

The complexity of the design, dynamic power consumption, and processing time are important factors in evaluating the performance of a microchip design [1]. Processing speed is inversely proportional to hardware complexity. We can see that the MAC unit, which uses multipliers of high complexity, will consume the most hardware resources, requiring up to 2376 transistors. On the other hand, with equivalent computational results, the XNOR-popcount design requires 1244 transistors when only using XNOR gates and an adder-tree, as shown in Table 1.

Table 1. *The performance comparison of the MAC design using multiplication and the XNOR-popcount method*

	Multipliers	XNOR-popcount	Savings
Transistor Count	2376	1244	48%
Power (µW)	122	38	69%
Delay (pS)	581	293	50%

The power consumption of the design includes static power and dynamic power components. Among them, dynamic power ($P_{dynamic}$) greatly affects the total power consumption of the design. Dynamic power is influenced by the supply voltage (V_{DD}), capacitance (C), and operating frequency (f) [14], and is detailed by the following equation:

$$P_{dynamic} = CV_{DD}^2 f \quad (3)$$

Based on the above equation, it can be observed that changes in the operating voltage (VDD) will significantly alter power consumption. This study simulates and evaluates designs at various operating voltage ranges such as 0.8V, 1V, and 1.2V to assess the impact of operating voltage on processing time and power consumption.

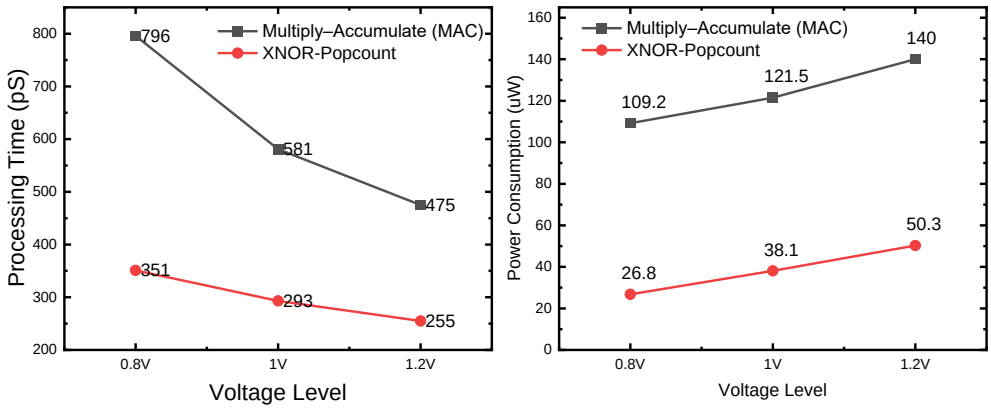

Figure 11. *Compare the operation of convolution on binary weights using MAC and approximate computation with XNOR-popcount a) Processing time b) Power consumption*

Compared to designs using conventional multipliers, the analysis results show that the MAC design has a higher latency and consumes more energy than the two XNOR-popcount designs. The XNOR-popcount design allows for high performance and lower power consumption. The operations and components optimized in the new design have helped improve performance and save energy in practical applications. Optimizing the XNOR-popcount design is very necessary, especially in embedded IoT systems, where most computer designs are integrated into intelligent and distributed systems at various locations, using limited energy sources from batteries or renewable energy sources. The XNOR-popcount design has reduced the number of transistors used by more than 48%, maximally increased the processing speed by two times, and the power consumption has been reduced by nearly 70% compared to designs using conventional multipliers as shown in Figure 11.

4. Conclusions

The XNOR-popcount, a hardware solution in binarized neural networks, is designed to replace the traditional multiply-accumulate (MAC) method that employs complex multipliers. The XNOR-popcount design helps to optimize the design area, lower power usage, and boost processing speed. This research carries out and assesses the performance of the XNOR-popcount design at the transistor level using Cadence software and 90nm CMOS technology. Simulation results show that for the same computational task, if the MAC operation employs XNOR-popcount, there can be a maximum reduction of up to 69%, 50%, and 48% in power consumption, processing time, and design complexity respectively, compared to the method that uses traditional multipliers. Therefore, the XNOR-popcount

design proves to be a beneficial approach for edge-computing platforms that have minimalist hardware design, limited memory space, and restricted power supply.

Acknowledgments

This work belongs to the project grant No: T2023-65 funded by Ho Chi Minh City University of Technology and Education, Vietnam.

Conflicts of Interest

The authors declare that they have no competing interests.

REFERENCES

[1] E. Nurvitadhi *et al.*, "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?" in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 5-14.
[2] J. Chen and X. Ran. "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, 1655-1674, 2019.
[3] Y. L. Cun, Y. Bengio, and G. Hinton. "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.
[4] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, 2014, pp. 10-14.
[5] L. Lai, N. Suda, and V. Chandra. "Deep convolutional neural network inference with floating-point weights and fixed-point activations," arXiv preprint arXiv:1703.03073, 2017.
[6] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 4820-4828.
[7] K. Hwang, and W. Sung. "Fixed-point feedforward deep neural network design using weights+ 1, 0, and− 1," in *IEEE Workshop on Signal Processing Systems (SiPS)*, 2014, pp. 1-6.
[8] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," *European conference on computer vision*, pp. 525-542, 2016.
[9] M. Courbariaux, I. Hubara, D. Soudry, R. E. Yaniv, and Y. Bengio. "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," arXiv preprint arXiv:1602.02830, 2016.
[10] K. A. Asha and K. D. Shinde, "Performance analysis and implementation of array multiplier using various full adder designs for DSP applications: A VLSI based approach," *Intelligent Systems Technologies and Applications*, pp. 731-742, 2016.
[11] A. Nigam and R. Singh, "Comparative Analysis of 28T Full adder with 14T Full adder using 180nm," *International Journal of Engineering Science Advance Research*, vol. 2, no. 1, pp. 27-32, 2016.
[12] S. Pandey, A. A. Khan, and R. Sarma. "Comparative analysis of carry select adder using 8T and 10T full adder cells," in *International Conference on Communication and Signal Processing*, 2014, pp. 985-989.
[13] H. Naseri and S. Timarchi, "Low-power and fast full adder by exploring new XOR and XNOR gates," *Transactions on very large scale integration (VLSI) systems*, vol. 26, no. 8, pp. 1481-1493, 2018.
[14] S. Vaidya and D. Dandekar. "Delay-power performance comparison of multipliers in VLSI circuit design," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 2, no. 4, pp. 47-56, 2010.

Pham Van Khoa received his B.S. and M. S. E. E. degrees in Computer Technology and Electronics Engineering from the University of Technology and Education, HCM City, Vietnam, in 2010 and 2014, respectively. In 2019, he obtained his Ph.D. in Electronics Engineering from Kookmin University (K.M.U.) in Seoul, Korea. In 2010, he joined the Integrated Circuit Design Research and Education Center (I.C.D.R.E.C.), contributing to developing VN8-01 MCU, the first commercially designed and fabricated microcontroller in Vietnam. From May 2011 to 2021, he was a member of the Faculty of Electrical and Electronics Engineering at Technology and Education, HCM City, Vietnam (H.C.M.U.T.E.), and currently holds the position of senior lecturer in the Department of Computer and Communication Engineering. Presently, he serves as the Head of Computer Technology Engineering at the Faculty of International Education, H.C.M.U.T.E. His research interests encompass low-power VLSI, memory design, Internet-of-Things (IoT) and power I.C. design. He has published research papers in a variety of prestigious journals, conferences, such as, Electronics Letters, IEEE Transactions on Nanotechnology, Journal of Semiconductor Technology and Science, Micromachines, International Journal of Computing, Indonesian Journal of Electrical Engineering and Computer Science, IEEE International Symposium on Circuits and Systems (ISCAS). Email: khoapv@hcmute.edu.vn. ORCID: https://orcid.org/0000-0002-6129-5856

Le Lai received his B.S. degrees in Computer Technology from the University of Technology and Education, Hochiminh City, Vietnam, in 2023. His research interests power I.C. design. Phone: 0585-856-578. Email: laile.science@gmail.com. ORCID: https://orcid.org/0009-0002-9434-0374

Tran Thi Thanh Kieu obtained her Master's degree in TESOL from the University of Southern Queensland in 2015. She has been a full-time lecturer at the Faculty of Foreign Languages at Ho Chi Minh City University of Technology and Education for over twelve years. During her time there, she has taught a variety of English courses for both majors and non-majors, including general English, English for Academic Purposes (EAP), with a focus on reading and writing skills, and exam preparation. Her research interests includes English linguistics, academic writing, and professional development for teachers. Email: kieuttt@hcmute.edu.vn. ORCID: https://orcid.org/0009-0004-7561-6122