Design of Precision Configurable Multiply Accumulate Unit for Neural Network Accelerator

Jian Chen, Xinru Zhou, Xinhe Li, Lijie Wang, Shengli Lu, Hao Liu†

School of Electronic Science and Engineering, Southeast University, Nanjing, Jiangsu China,

Abstract

With the increasing size of models in Convolutional Neural Networks (CNNs) recently, the demand for memory size, bandwidth and computational resources has gradually become a central issue. Quantification has a pivotal role in dramatically reducing the computation of CNN models and bandwidth. However, quantization technology is difficult to improve the throughput and power efficiency of accurately fixed accelerators. Different applications have different requirements for accelerators in all aspects, and accurately fixing accelerators lacks the flexibility to meet these requirements. In this paper, a precision configurable processing unit (PE) is proposed, which not only simplifies the computing unit and the external complex configurable logic, but also introduces the concept of approximate calculation, while ensuring a certain precision of CNN. For the first time, approximate computation is introduced in a configurable computational unit, which allows the architecture to further reduce power consumption based on bit-level flexibility and to accommodate parameters from different quantization methods of the network. The design of this paper is implemented in SMIC 40nm process library. Compared with Bit Fusion [1], this method achieves the lowest accuracy of 98.49% in Lenet, ensuring that the area and power consumption are reduced by 53.2% and 19.8% respectively.

Keywords

Convolutional Neural Networks; Precision Scaling; Quantization; Approximate Calculation

1 Introduction

CNNs have achieved great success in many computer vision tasks such as image recognition [2-4] and target recognition [5-8]. In the recent development of CNNs, the increasing model size has led to significant demands on memory size, bandwidth and computational resources [7,8].

To address these issues, many model compression methods such as pruning [1,9] and quantization [5-8] have been proposed to reduce the storage and computational requirements of CNNs. Quantization can significantly reduce the size of CNN models and alleviate the memory-intensive problem, which is beneficial to reduce the bandwidth requirements [6]. However, most of the current accelerators fail to utilize quantization models to solve the computationally-intensive problem[6]. Most accelerators [14,15] perform multiply accumulate (MAC) operations with fixed high precision, but many MAC operations with quantization are not necessary for such high precision [6]. Quantization techniques are difficult to improve the throughput rate and power efficiency of the precision-fixed accelerators. Different applications have different requirements for accelerators in various aspects, and precision-fixed accelerators lack the flexibility to meet these requirements.

Therefore, many precision-configurable CNN accelerators have been recently proposed [10-13], where activations and weights can be partially or fully scaled. For example, the Dynamic Voltage, Accuracy, and Frequency Scaling (DVAFS) [10], first proposed by Bert Moons et al., is based on the data gating approach and reuses full adder units that are not effective at scaled accuracy, which allows

AIoTC2022@International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology
EMAIL: 220206021@seu.edu.cn (Jian Chen), 220216012@seu.edu.cn (Xinru Zhou), 220216105@seu.edu.cn (Xinhe Li), 220205783@seu.edu.cn (Lijie Wang), lsl@seu.edu.cn (Shengli Lu), nicky_lh@seu.edu.cn (Hao Liu)

both activations and weights to be scaled in proportion. Compared with the traditional data gating approach, DVAFS, with a shortened critical path and dynamic adjustment of the clock frequency, adopts the sparsity of convolution in a dedicated processor architecture during the chip implementation and achieves variable voltage and frequency with accuracy.

However, with the increase of performance, the lower precision components require complex configurable logic. The decrease in precision also requires more activations and weights to perform the computation of precision-configurable units such as Bit-Fusion [13] and BitBlade[16]. The increase in activation and weight requirements also increases the demand for bandwidth and logic resources, which additionally leads to an increase in power consumption and a reduction in hardware utilization, reducing the benefits of quantification. This paper designs a precision configurable module and introduces an approximation method to try to reduce power consumption and improve hardware utilization.

The main contributions of this paper are as follows: (1) A precision-configurable computational unit is proposed to simplify the computational unit and the complex external configurable logic under the premise of a certain accuracy of the neural network; (2) For the first time, an approximation is introduced in the configurable unit, which enables the architecture to further reduce power consumption on the basis of bit-level flexibility and to adapt to parameters from multiple quantization methods of the network; (3) The design of this paper is implemented in SMIC 40nm process library. Compared with Bit Fusion [1], this method achieves the lowest accuracy of 98.49% in Lenet, ensuring that the area and power consumption are reduced by 53.2% and 19.8% respectively.

The remainder of this paper is organized as follows. The second section introduces the background work of Quantization Compression and precision scalability. The third section analyzes the hardware design content and main innovations, and the fourth section evaluates the performance of the whole design. Finally, the fifth section summarizes the full text.

2 Related Work

When CNNs are applied in embedded devices, it is necessary to consider not only the demand on memory size, bandwidth and computational resources brought by the huge computational volume, but also the problem of limited energy supply. Quantization [5-8] is a method used to reduce the storage and computation of CNNs. Although quantization brings some accuracy loss, its impact on accuracy loss is negligible.

MAC operations account for 99% of the total operations in CNN. 97.3% of MAC operations can be performed at less than 4 bits without affecting the accuracy, and even most of the operations can be done at 1 bit. [1] Since the number of multiplication operations is proportional to the product of operand bit widths, the quantified network is able to speed up significantly. As the bit widths of activation and weight are reduced, the number of bits to access the memory is reduced and thus the power consumption to access the memory is also reduced. However, quantization techniques are difficult to apply to DNN accelerators with fixed bit widths to improve their throughput and energy efficiency, so it is especially significant to design MACs that can dynamically adapt to the bit widths of operands.

Accuracy-scaling MACs can adapt to the input parameters of different quantified network, which makes the hardware much more flexible. Precision-scaling MACs are efficiently parallelized or serialized. Data gating is first proposed in configurable arithmetic circuits, after which Subword Parallelism, Divide and Conquer, and Bit-serial architectures were proposed, respectively. Bit-Fusion [1] is a 2D precision-scaling method based on Divide-and-Conquer. This method computes and communicates with fine-grained as possible without loss of precision, and reduce the power consumption of access memory while increasing the on-chip storage capacity by reducing the total number of bits of on-chip and off-chip memory.

Precision-scaling units are generally composed of adders, multipliers and external configuration units. The research on adders and multipliers is very mature, and we fuse the external configuration unit with the MAC unit in order to reduce the overall area and power consumption. The basic principle of calculating multiplication in this design is the same as that of Bit-Fusion. However, this design takes into account the fact that the partial sums of high bits and low bits do not affect each other and can be

added by bit splicing. Thus, the use of bit splicing in our design reduces the use of accumulator and thus reduces the area overhead. In addition, in this design, there is no need to make up 1-bit sign bit after splitting into lower bits. The smallest unit of this design implements a 2-bit multiplication operation, which further saves area overhead compared to Bit-Fusion which implements a 3-bit multiplication operation.

In addition to the above optimization methods, this paper also introduces approximate means in configurable computing units for the first time. We use the LOA adder to approximate optimize the configurable computing unit, and further reduce the area and power consumption on the premise of ensuring accuracy.

3 Proposed Design

The main purpose of this design is to make the architecture have bit level flexibility on the basis of reducing power consumption, and finally be able to adapt to the parameters from various quantization methods of the network. The core of this design is the dynamic implementation of operand bit-width adjustment with a multiplexer to select the bit-width mode. The configurable MAC architecture is able to dynamically implement calculations for three cases--8×8, 4×4, and 2×2, which is sufficient for application to neural networks and avoids fine-grained calculations.

3.1 Throughput Analysis

As the variety of computations supported by MAC increases, the complexity of the hardware design increases. Because fine-grained computations can lead to complex architectures, the required granularity needs to be carefully chosen. If two computation cases have similar accuracy, the one with better throughput can be used instead of the other, which reduces the variety of computations and simplifies the hardware design.

In our design, the configurable MAC is applied to the LeNet5 network with the activation and weight quantified to 8 bit, 4 bit and 2 bit, respectively. Since the quantization of activation affects the accuracy more than the quantization of weight, only the six computation cases-- 8×8, 8×4, 8×2, 4×4, 4×2, and 2×2 --shown in Table II are considered. As shown in Table I, there is little difference in accuracy between these six computation cases.

TABLE I. TOP-1 ACCURACY FOR VARIOUS COMPUTATIONAL CASES IN IMAGENET [6]

Computational cases	***Alex-Net***	***VGG16***	***Res18***	***Res34***	***Res50***
8×8	54.5	71.1	69.6	73.6	76.2
8×4	54.2	70.1	70.1	73.1	74.7
8×2	50.2	N/A	67.6	71.5	72.8
4×4	54.4	70.5	67.0	N/A	73.8
4×2	50.5	N/A	N/A	N/A	N/A
2×2	51.3	69.1	67.0	N/A	74.2

TABLE II. THROUGHPUT OF BITBLADE IN DIFFERENT COMPUTING SITUATIONS [6]

Throughput	***2×2***	***4×2***	***4×4***	***8×2***	***8×4***	***8×8***
VGG16	4.33	2.54	1.38	1.38	0.71	0.36
ResNe-t152	3.67	2.31	1.30	1.30	0.69	0.35

The effect of simultaneous quantization of weights and activation on the training results was investigated on the PyTorch platform to verify the feasibility of the simplified computational cases. It is evident from Fig.1 that the quantization of weights and activation has little effect on the output accuracy of the trained model. The activation is more sensitive to changes in the number of quantization bits due to a larger range of activation quantization errors. Therefore, the accuracy

configurable MAC unit supports 8×8, 4×4, and 2×2 computations, which can greatly reduce the complexity of hardware computation within the accuracy loss allowed.

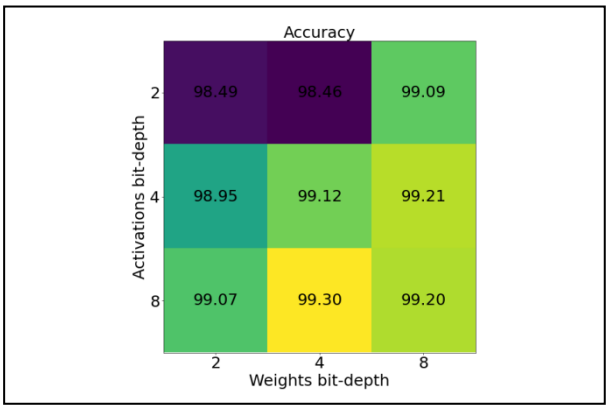

Figure1. The relationship between quantification of weights, activation and accuracy in LeNet-5

3.2 Configurable unit

1)Using a 2-bit multiplier based on multiplexer: The smallest computation unit of this design is the bit-level processing element, which is capable of 2-bit multiplication. When this design perform the 2-bit multiplication, the multiplexer determines the operand with or without sign, avoiding the addition of sign bits in Bit-Fusion.

A and B are two signed/unsigned numbers. The inputs and outputs of the multipliers are in the form of the complement of signed numbers. Cond(1) represents the case of "unsigned A × unsigned B"; cond(2) represents the "signed A × unsigned B"

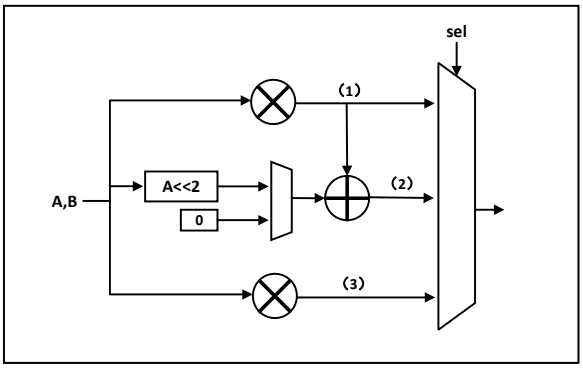

Figure2. 2-bit multipiler design of this paper, note: In later papers we call this 2BM

$$\begin{aligned} cond(1): A\times B &= (2A[1]+A[0])\times(2B[1]+B[0]) \\ &= 4A[1]B[1]+2A[1]B[0]+2A[0]B[1]+A[0]B[0]. \end{aligned} \quad (1)$$

$$\begin{aligned} cond(2): A\times B &= (-2A[1]+A[0])\times(2B[1]+B[0]) \\ &= -4A[1]B[1]-2A[1]B[0]+2A[0]B[1]+A[0]B[0] \\ &= cond(1)+([2B[1]+B[0]]<<2)\times A[1]. \end{aligned} \quad (2)$$

$$\begin{aligned} cond(3): A\times B &= (-2A[1]+A[0])\times(-2B[1]+B[0]) \\ &= 4A[1]B[1]-2A[1]B[0]-2A[0]B[1]+A[0]B[0]. \end{aligned} \quad (3)$$

2)Adopting bit-splicing: Bit-splicing method directly merges partial products that do not interfere with each other. Taking the 4×4 shown in Fig. 3 as an example, the 4-bit A and B are first split into 2-bit A[3:2], A[1:0], B[3:2] and B[1:0]. The product A[1:0]×B[1:0] of the lower 2-bit and the product

A[3:2]×B[3:2] of the higher 2-bit do not interfere with each other and can be directly bit-spliced without going through accumulation. In the 4-bit multiplication, a bit-splicer is used instead of an adder and a shifter to avoid shifting and accumulation of the higher four bits of the partial sum.

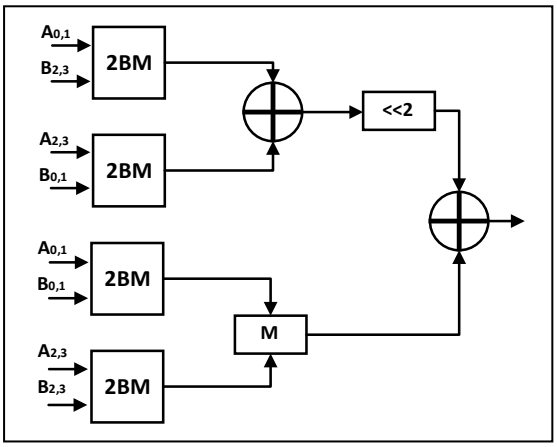

Figure3. 4*4 multiplier of bit-splicing method

As shown in Table III, the number of multipliers, shifters, and adders required by the bit-splicing-based approach proposed in this paper is significantly less than that required in Bit-Fusion, and this advantage becomes more and more obvious as the number of bits of the computed multipliers becomes larger. In comparison, this design have smaller area and lower power consumption, so the overall area and power consumption of the configurable MAC are smaller.

TABLE III. COMPARISON BETWEEN BIT-FUSION AND BIT-SPLICING FOR MULTIPLICATION

	Bit-Fusion			***This design***		
	2×2	***4×4***	***8×8***	***2×2***	***4×4***	***8×8***
Minimum multiplier	3-bit signed multiplier			2-bit multiplier based on multiplexer		
Number of multipliers	1	4	16	1	4	16
Number of shifters	0	3	15	0	1	9
Number of adders	0	3	15	0	2	8

3)Building configurable multipliers: The 2BM which perform 2-bit multiplication are arranged in the space. As shown in Fig. 4, a complete configurable MAC is composed of 16 2BMs and is capable of accommodating MAC operations of 2bit, 4bit, and 8bit DNN layers.

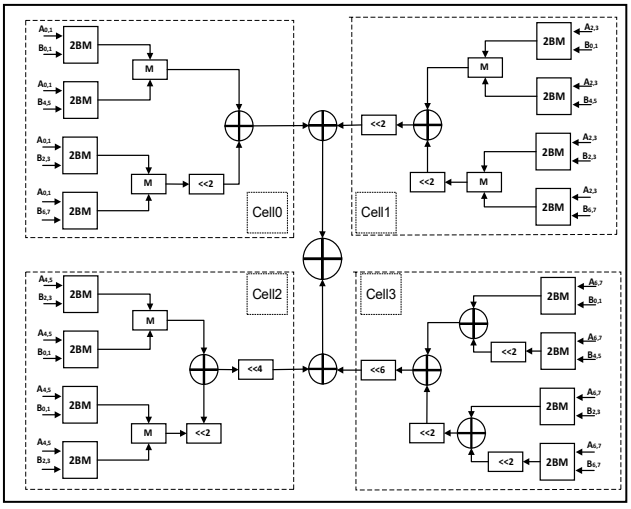

Figure4. The architecture of configurable MAC

The steps to implement the multiplication of two 8-bit signed numbers A and B in a configurable MAC are as follows:

Step1: Split the two multipliers A,B into four 2-bit numbers respectively -- A[1:0] (denoted as $A_{0,1}$), A[3: 2] (denoted as $A_{2,3}$), A[5:4] (denoted as $A_{4,5}$), A[7:6] (denoted as $A_{6,7}$), B[1:0] (denoted as $B_{0,1}$), B[3:2] (denoted as $B_{2,3}$), B[5:4] (denoted as $B_{4,5}$), and B[7:6] (denoted as $B_{6,7}$).

Step2: $A_{0,1}$, $A_{2,3}$, $A_{4,5}$, and $A_{6,7}$ are broadcast to each 2BM of four cells, respectively.

Step3: Each cell receives the complete bits of B and assigns them to each 2BM accordingly.

Step4: The partial products obtained from the four cells are shifted accordingly and then added up to obtain the product of A and B.

4)Configuring output bandwidth mode: For different configuration modes, the input bandwidth is 8 bits, but the corresponding output bandwidth varies greatly in different configuration modes as shown in Table IV. To reduce the huge pressure on the output bandwidth, the bandwidths in different modes are reused. In both the 8×8 and 4×4 input modes, the output bandwidth is reused to the 2×2 mode, and the final overall output bandwidth is 64 bits, thus reducing the area and power loss due to bandwidth.

TABLE IV. INPUT AND OUTPUT BANDWIDTH FOR DIFFERENT MODES

Input Mode (bit×bit)	***Input bandwidth(bit)***	***Output bandwidth(bit)***
2×2	8	64
4×4		32
8×8		16

3.3 Approximate calculation unit

In this experiment, we perform a hardware implementation of the above proposed model and a simple analysis of the whole implementation is performed in Design Compiler. The area of the whole design and the percentage of the adder module, 2BMs and the external configuration module are counted in Table V, and it is found that the adder area accounts for a larger percentage. Due to the fault tolerance of CNNs, the adder is approximated to operate with a certain accuracy.

TABLE V. PERCENTAGE OF AREA OF EACH PART IN THE HARDWARE IMPLEMENTATION

	this design	***Adder***	***2BM***	***External configuration module***
Area(um^2)	1471.83	796.41	511.9	163.52
Percentage (%)	100	54.1	34.8	11.1

Approximate adders are mainly classified as Accuracy Configurable Adder (ACA), Speculative Carry Select Addition (SCSA), Carry-Skip Adder (CSA), Error-Tolerant Adder (ETA) and Low or Adder (LOA). The comparison of various approximate adders is as shown in Table VI. It is found that LOA has the smallest area and power consumption due to the complete use of logic or gates for low-bit operation, but it has the highest error rate because the accuracy is not considered. Since adders have different requirements for different bit widths, we finally choose LOA.

TABLE VI. COMPARISON OF APPROXIMATE ADDERS [17]

Types of adders	***Area (um^2)***	***Delay (ns)***	***Power (uW)***	***Error rate (%)***	***Mean Relative Error (um^2)***
LOA	53.2	0.39	65.9	89.99	1.0
ETAII	71.6	0.55	80.6	5.85/16.94	2.6
ACA	73.8	0.25	118.4	16.66/16.34	18.9
SCSA	109.2	0.32	134.5	5.85	2.6
CSA	142.5	0.39	97.8	0.18/0.91	0.15

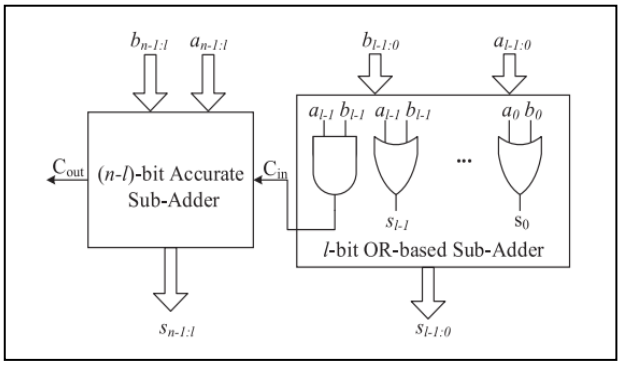

Figure5. LOA schematic

The approximate bit widths required for different bit-width adders are different. We modeled each adder in Matlab, selected certain equal intervals, and tested its MRED using Monte Carlo method. Because the MRED of the computational unit is usually required to be less than 5%[18], we finally determined the approximate bit-width to be 2/3/4/6 for bit-width 6/8/10/16 bits. We list three cases for different bit widths in Table VII for comparison.

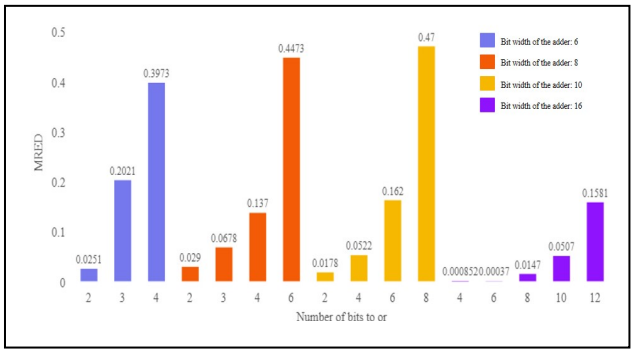

Figure6. Different bits of LOA and the corresponding MRED

TABLE VII. BIT-WIDTH SELECTION OF ADDERS AND OVERALL ACCURACY TESTING

Case	*Bit-width*	*Accuracy of each adder*	*MRED of MAC*
Case1	6	2	0.0025
	10	2	
	16	4	
Case2	6	2	0.0300
	10	4	
	16	6	
Case3	6	3	1.8396
	10	6	
	16	8	

After modeling the selected low approximation bits corresponding to different bit-width adders, the hardware implementation of each adder will be performed and the implemented hardware will be tested in Design Compiler, and the final test results are shown in Table VII. Finally, we use case 2 as the final bit width selection.

TABLE VIII. COMPARISON OF NUMBER OF ADDERS AND AREA

Bit-width	*Number*	*Area (um^2)*	*Total area (um^2)*	*Area of Approximation (um^2)*	*Total area of Approximation (um^2)*
6	6	30.88	185.28	23.46	140.76
8	4	40.93	163.72	29.45	117.80
10	4	50.99	203.96	35.43	141.72

Bit-width	Number	Area (um^2)	Total area (um^2)	Area of Approximation (um^2)	Total area of Approximation (um^2)
16	3	81.15	243.45	57.45	172.35

As shown in Figure VIII, we count the number of adders, the area of a single accurate and approximate adder, and the total area of a fixed bit width adder. The sum of the statistical exact adder area and the sum of the approximate post adder area are compared to achieve a gain of 39.41% in area, which is a huge gain for the entire computational unit.

4 Evaluation

Bit-Fusion [1], which was presented at ISCA in 2018, was selected for comparison. The MAC design of Bit-Fusion is to add a symbol bit to the original 2*2 multiplication unit, and finally build it into a minimum calculation unit of 3*3. For the case of 4*4 or 8*8, a shifter is used to replace the multiplication carry, and the results are combined and added. The design of this paper optimizes the external configuration module and multiplication in bit fusion to reduce the area and power consumption of the computing unit. The design of Bit-Fusion is realized under the 45nm process. This paper will realize and compare the Bit-Fusion design and this design under the same experimental conditions. The design is tested in Design Compiler using SMIC 40nm process library.

4.1 Performance Analysis

1)Comparison of 2-bit minimum multiplication: Bit-Fusion uses four full adders(FAs) and three half adders(HAs) while this design uses only one FAs, four HAs and two data selectors. The benefits of using this design are considerable as it reduces the area by 35.5% and the power consumption by 47.5% compared to the Bit-Fusion design.

2)Comparison of 8-bit multiplication: The 8-bit multiplier is compatible with sixteen multipliers with 2-bit input or four multipliers with 4-bit input. Due to the bit-selective design and the use of bit-splicing, the benefits of this design are considerable as it reduces the area by 53.1% and the power consumption by 40% compared to the Bit-Fusion design.

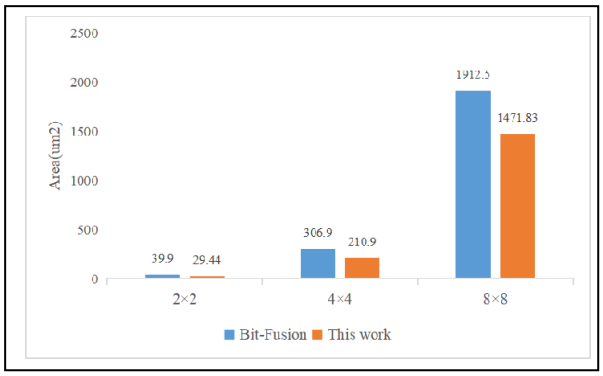

Figure7. Comparison of area

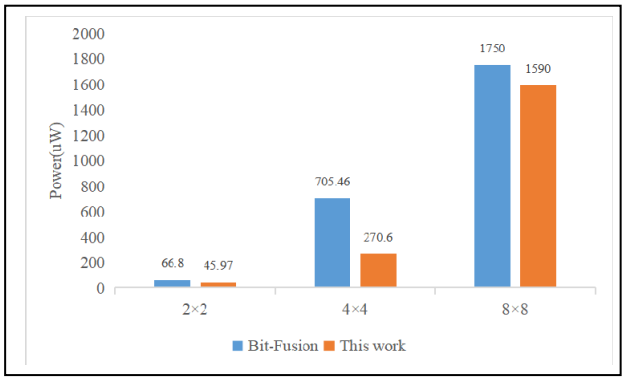

Figure8. Comparison of power consumption

3)Introduction of approximate adders: In this paper, we introduce approximate adders into precision-scaling schemes for the first time. For the selection of the approximation adder, we choose the LOA in this paper. After the hardware implementation, we test and compare with the exact design in Design Compiler, and the specific results are shown in Table IX, where our design achieve some gains in area and power consumption.

TABLE IX. AREA AND POWER CONSUMPTION IN PRECISE AND APPROXIMATE CASE

8-Bit (SMIC 40)	***Precise case***	***Approximate case***	***Comparison (%)***
Area(um^2)	1471.83	1247.99	17.9%
Power(mW)	1.59	1.46	8.9%

The final approximate design solution is compared with the comparative design Bit-Fusion, and the results are shown in Table X. The accuracy configurable unit is greater than 53.2% in power consumption and greater than 19.8% in area reduction.

TABLE X. AREA AND POWER CONSUMPTION OF OUR DESIGN AND BIT-FUSION IN APPROXIMATE CASE

8-Bit (SMIC 40)	***Bit-Fusion***	***this design***	***Comparison (%)***
Area(um^2)	1912.5	1247.99	53.2
Power(uW)	1.75	1.46	19.8

4.2 Accuracy Analysis

Table XI shows the approximate bit width corresponding to each adder. In this paper, after modeling the selected low approximation bits corresponding to different bit width input modes in MATLAB, the corresponding MRED is tested, and the final test results are shown in Table XII. Finally, we use the case in lenet and test the final accuracy. In each mode, we use 10000 pictures to test the final accuracy, as shown in table XII.

TABLE XI. BIT-WIDTH SELECTION OF ADDERS AND OVERALL ACCURACY TESTING

Bit-width	***Number***	***Accuracy of each adder***	***MRED of MAC***
6	6	2	0.030
8	4	3	
10	4	4	
16	3	6	

TABLE XII. MRED AND RECOGNITION ACCURACY UNDER DIFFERENT INPUT MODES

Input Mode	***MRED of MAC***	***Recognition Accuracy***
2	0	98.49%

Input Mode	*MRED of MAC*	*Recognition Accuracy*
4	0.019	99.12%
8	0.030	99.20%

5 Conclusion

Based on the fault-tolerance of CNNs, the accuracy-configurable unit enables the circuit to accept a variety of network parameters by means of adding additional configuration units. In this paper, from the perspective of improving the flexibility of accelerators, the precision- scaling MAC is designed to adapt to multiple network structures while ensuring low power consumption. The hardware performance of the accelerator will be improved, and the worst accuracy in Lenet will reach more than 98.49%. The precision-configurable cell carried out in SMIC 40nm process has a power gain of more than 53.2% and an area reduction gain of more than 19.8% compared to Bit-Fusion [1].

6 References

[1] H Sharma, J Park, N Suda, et al. Bit Fusion: Bit-Level Dynamically Composable Architecture for Accelerating Deep Neural Networks [M]. 2018 ACM/IEEE 45TH ANNUAL INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE (ISCA). 2018: 764-775.

[2] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS), 2016, pp. 2082–2090.

[3] X. Yang et al., "DNN dataflow choice is overrated," Sep. 2018, arXiv:1809.04070. [Online]. Available: https://arxiv.org/abs/1809.04070.

[4] V. Camus, M. Cacciotti, J. Schlachter, and C. Enz, "Design of approximate circuits by fabrication of false timing paths: The carry cutback adder," IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 8, no. 4, pp. 746–757, Dec. 2018.

[5] D D Lin, S S Talathi, V S Annapureddy. Fixed Point Quantization of Deep Convolutional Networks [M]. INTERNATIONAL CONFERENCE ON MACHINE LEARNING, VOL 48. 2016.

[6] W J Liu, J Lin, Z F Wang. A Precision-Scalable Energy-Efficient Convolutional Neural Network Accelerator[J]. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I-REGULAR PAPERS, 2020, 67(10): 3484-3497.

[7] B Jacob, S Kligys, B Chen, et al. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference [M]. 2018 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). 2018: 2704-2713.

[8] J Wu, C Leng, Y Wang, et al. Quantized Convolutional Neural Networks for Mobile Devices [M]. 2016 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). 2016: 4820-4828.

[9] Baugh, C. R., and B. A. Wooley, "A Two's Complement Parallel Array MultiplicationAlgorithm,"IEEE Trans. Computers, Vol. 22, pp. 1045–1047, 1973.

[10] J Albericio, P Judd, T Hetherington, et al. Cnvlutin: Ineffectual-Neuron-Free Deep Neural Network Computing [M]. 2016 ACM/IEEE 43RD ANNUAL INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE (ISCA). 2016: 1-13.

[11] A Parashar, M Rhu, A Mukkara, et al. SCNN: An Accelerator for Compressed-sparse Convolutional Neural Networks [M]. 44TH ANNUAL INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE (ISCA 2017). 2017: 27-40.

[12] X Zhou, Z Du, Q Guo, et al. Cambricon-S: Addressing Irregularity in Sparse Neural Networks through A Cooperative Software/Hardware Approach [M]. 2018 51ST ANNUAL IEEE/ACM INTERNATIONAL SYMPOSIUM ON MICROARCHITECTURE (MICRO). 2018: 15-28.

[13] S Zhang, Z Du, L Zhang, et al. Cambricon-X: An accelerator for sparse neural networks; proceedings of the 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), F 15-19 Oct. 2016, 2016 [C].

[14] T Luo, S Liu, L Li, et al. DaDianNao: A Neural Network Supercomputer[J]. IEEE TRANSACTIONS ON COMPUTERS, 2017, 66(1): 73-88.
[15] Y-H Chen, T Krishna, J S Emer, et al. Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks[J]. IEEE JOURNAL OF SOLID-STATE CIRCUITS, 2017, 52(1): 127-138.
[16] S Ryu, H Kim, W Yi, et al. BitBlade: Area and Energy-Efficient Precision-Scalable Neural Network Accelerator with Bitwise Summation; proceedings of the 2019 56th ACM/IEEE Design Automation Conference (DAC), F 2-6 June 2019, 2019 [C].
[17] Jiang,H.,Liu,C., Liu,L. et al.(2017) A Review. Classification, and Comparative Evaluation of Approximate Arit-metic Circuits.ACMJournal onEmerging Technologies in Computing Systems(JETC),13,1-34.
[18] Q. Li, X. Fan, J. Chen, H. Li and H. Liu, "A Hardware Efficient Approximate Shift Multiplier with High Accuracy," 2021 IEEE 14th International Conference on ASIC (ASICON), 2021, pp. 1-5, doi: 10.1109/ASICON52560.2021.9620363.