Time-Domain Multiply Accumulator Circuits for CNN Processors in 28 nm CMOS Technology

Thesis

Submitted to University College Dublin in part fulfilment of the requirements for the degree of Master of Engineering in Electronic and Computer Engineering

Xutong Wu

Student Number: 17211083

Supervisor: Dr. Teerachot Siriburanon

September 2019

ABSTRACT

A time-domain Multiply Accumulator (MAC) array is proposed and designed in 28 nm CMOS technology for convolutional operation between matrixes which is the fundamental operation of the art-of-state convolutional neural network (CNN) models. The multiply-and-accumulate operations perform different time delays using two capacitor arrays to control the delay generation. Compared with other analog MAC propositions, the time-domain MAC array performs better on frequency of operations which can achieve higher than 50MHz. Forcing on the signal MAC unit, the proposed 6-bit time-domain MAC achieves 0.1 GOPS ideal throughput with 50MHz and consumes 120 μW. The range of the time delays and the linearity of the results are discussed in thesis.

Key Word: Time-domain, Multiply Accumulator, Neural Network, Digital-to-time Converter

ACKNOWLEDGEMENT

I would like to thank my supervisor Dr. Teerachot who gave me a great deal of support and assistance during the whole time of the project. I would like to thank members in research group who helped me and gave me advice. I would like to thank my parents for their support. In addition, I would like to thank to my future pet: a black cat.

Contents

1 Introduction and Background

Artificial intelligence (AI) has gradually appeared in everywhere of people's lives, from intelligent assistants to autonomous vehicles. AI technologies enable computers to have the ability to learn, analyze and create from data sets. Convolutional Neural Network (CNN) is the main representative of the breakthrough in artificial intelligence today. By applying CNN models, the computer has reached nearly human level in image classification, speech recognition, natural language processing [1; 2].

In order to cope the demand of huge computational resources of CNN models, the common solution is deploying CNN models in the servers or 'cloud'. When processing the CNN models, the mobile devices sends the input data to the 'cloud' and receives results from the 'cloud' after calculation [3]. However, based on cloud-based solutions, transmitting the large amount of data (videos, images) results in a larger delay and higher power consumption of mobile devices. During the data transmission, the information security issues are also worth considering.

To deploy CNN models to the edge devices becomes a new re-

search direction which allows mobile devices processing CNN models locally. The CNN accelerating chip is one of the solutions to reach requirements in processing speed, power consumption and hardware area[4]. Comparing the large computation resources required by using 32-bit floating point numbers to run CNN models, there have been recent works in reducing precision for running CNN models in energy-efficient. For example, Vanhoucke et al.[5] show that the CNN models can robustly work in 8-bit fixed point. Courbariaux et al.[6]show that the resolutions for simple image classification tasks (e.g. Handwritten recognition) can go down to 1 bit. The ultra-low precision solutions bring the opportunity to use analog circuit to replace digital circuit in processing computational operations.

The fundamental unit for supporting CNN models is Multiply Accumulator (MAC) which consist two basic operation units: multiplier and adder[7]. As CMOS technology scales, more computational power can be achieved. To further improve energy efficiency of traditional standard digital operation units [8; 9], the analog mixed-signal design in advanced CMOS technology can help improving performance such as power consump-

tion, operation speed and precision requirement. For example, Switched-capacitor based, current-mode and voltage-based approaches[10; 11; 12] allow MAC work in low power consumption or be built in small area. However, these propositions are limited by the voltage range or frequency range.

On the contrary, in time domain, the digital numbers are converted to time delay by using pulse generation structure, where is no limited for voltage or current, and have potential to achieve ultra-high frequency which means a higher operation speed. For example, the time-domain design proposed by Sayal el al.[13], which is working with binarized weight (1 bit) with ultra-low energy and 0.8 GHz frequency. Everson et al.[14] came up 2X time amplifiers based structure in maximum 3 bit precision with 1.7GHz. Although these designs achieve good performance in Handwritten recognition tasks, for applying the CNN chip in complex tasks (e.g. autonomous vehicles), a higher precision is required.

This thesis researches the possibility to build the analog MAC based on time domain with 6-bit precision. The specific research

aims of this project are to:

- Understand the basic algorithm of the Convolutional Neural Network, especially the data flow and the unit operation.
- Compare different art-of-state CNN chips in different techniques and figure out the advantages of designing CNN chips in analog circuit, especially in time domain.
- Design the top-level structure of the time-domain MAC array. Focus on the single MAC design and verify the circuit by simulating in software.

2 Literature Review

2.1 CNN Basics

The CNN algorithm is constructed by stacking multiple computation layers for feature extraction and classification[15]. Each layer is a set of nonlinear functions of weighted sums at different coordinates of spatially nearby subsets of outputs from the prior layer[9]. Normally, the deeper (more layers) CNN lead to better accuracy of results. Meanwhile, more hidden layers also brings a large number of computational resources requirements. For example, Figure 1 shows the structure of the 16 layers 'VGG' model[16], which is one of the common CNN models that secured the second place in the image classification task of ILSVRC-2914 challenge with 7.3% test error[17]. This model requires more than 15 Giga Floating-point operations per classification[18].

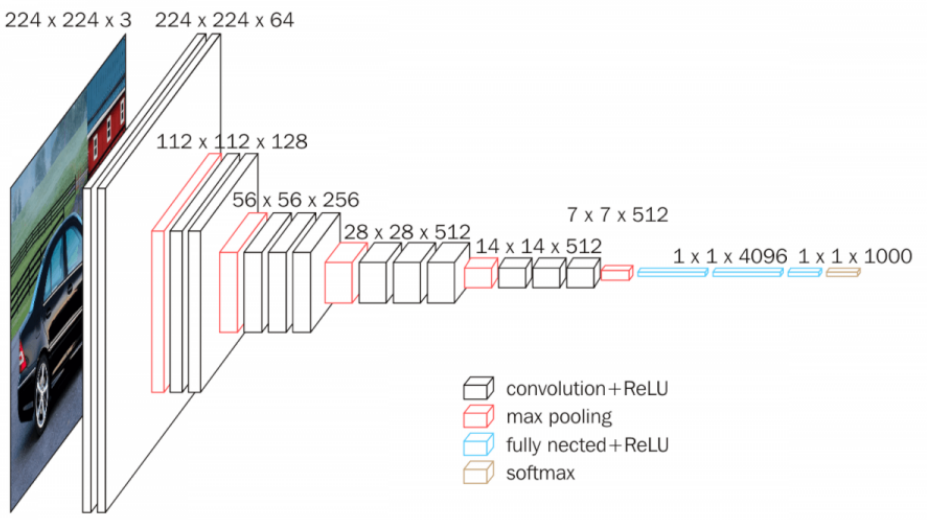

Figure 1: The structure of the VGG-16 model [16]

The most computational resource hungry part in CNN models is the convolutional layer shown in Figure 2. The number of operations for a M filters, C channels convolutional layer can be calculated by the following equation:

$$2 * (Filter\ size)^2 * (Input\ size - Filter\ size + 1)^2 * M * C \quad (1)$$

where in normally, the *Filter size* is chosen from 2 to 11, the *Input size* is chosen from 28 to 32, C and M could up to more than 100.

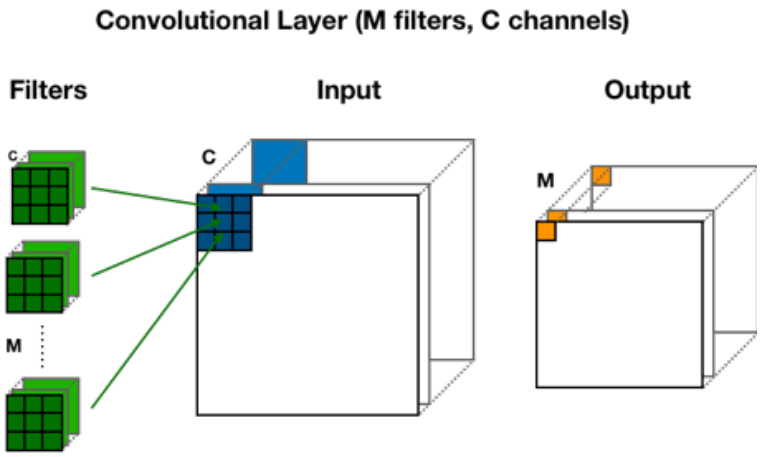

Figure 2: Convolutional Layer of the CNN mode

The convolutional layer can be refined to the numerous convolutional calculations between two matrix shown in Figure 3 (a). Moreover, the fundamental operation of the convolutional layer is multiplication operations and addition operation shown in Figure 3 (b) which can be expressed as the equation (2).

$$\sum Weight * Input = Output \tag{2}$$

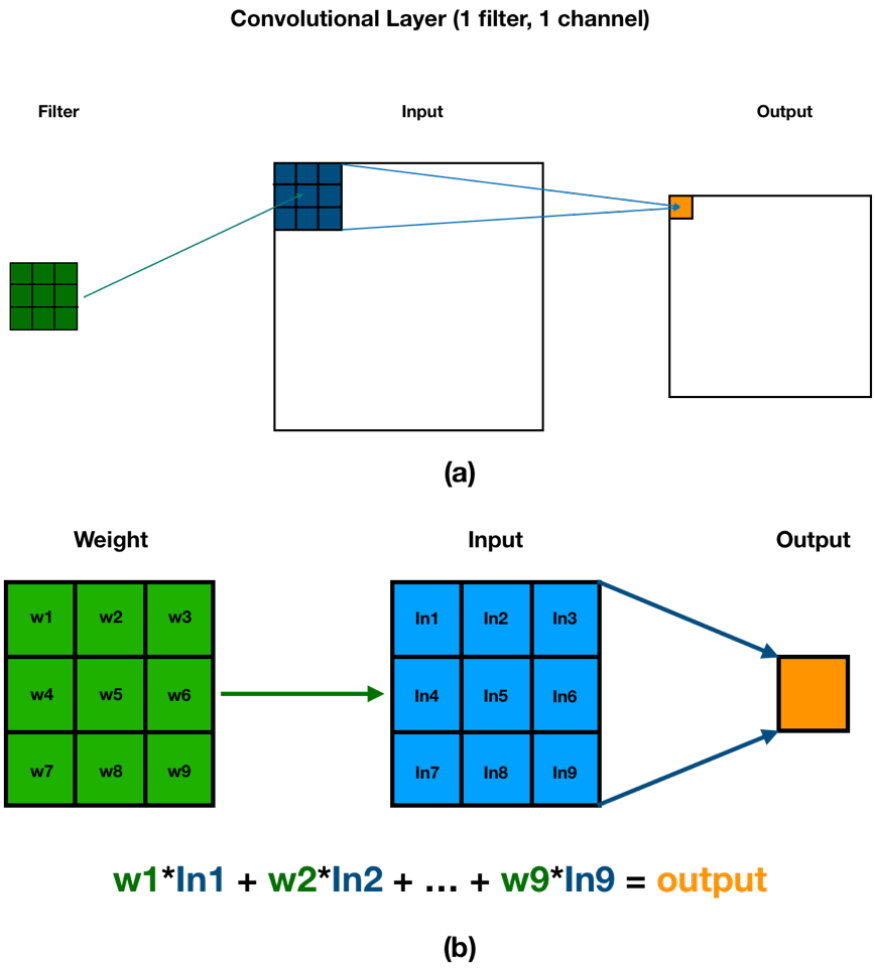

Figure 3: The fundamental operations of the convolutional layer

2.2 Multiply Accumulator (MAC)

For processing the convolutional operation between two matrix, the MAC array is considered to use. The Input and Weight are sent to the MAC array in parallel and the output is received from the last MAC. The diagram of the MAC array is shown in

Figure 4. There is a choice to build the MAC array in digital domain or analog domain.

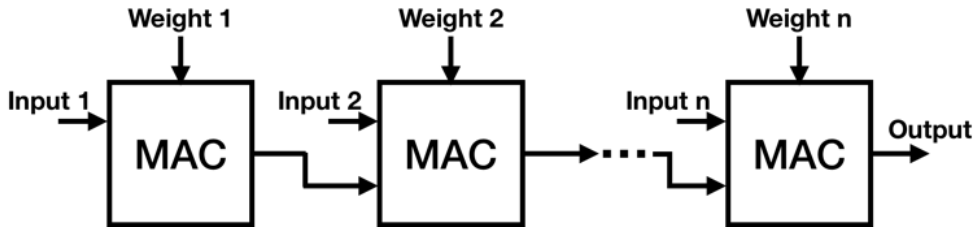

Figure 4: The structure of MAC array

2.2.1 Digital MAC

Because of Input and Weight are digital numbers, the digital MAC array is widely used in CNN chips. Figure 5 shows the basic structure of the digital MAC which can perform calculations in high precision.

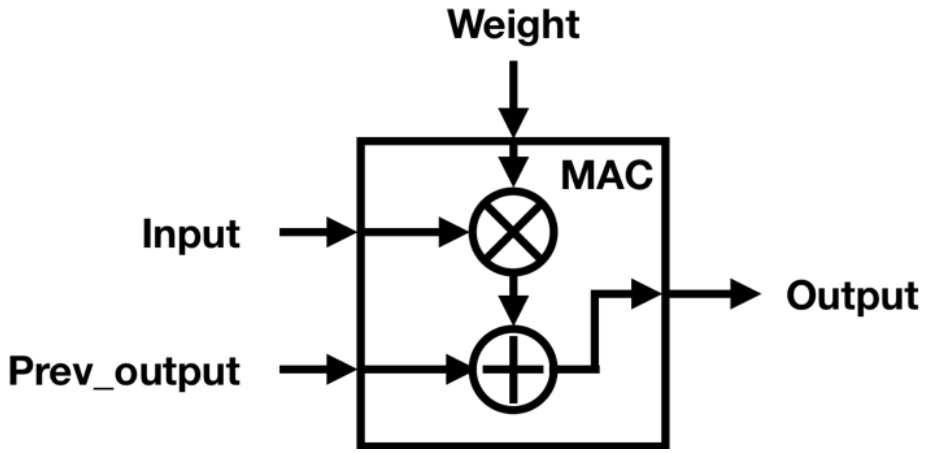

Figure 5: The structure of digital MAC

However, it is hard to modify and improve the multiplier or the adder in the digital MAC. The common way to improve the performance of the digital CNN chip is to change the data flow of the digital MAC array to reduce the reading and writing times between MAC array and memory to reduce the power consumption as shown in Figure 6.

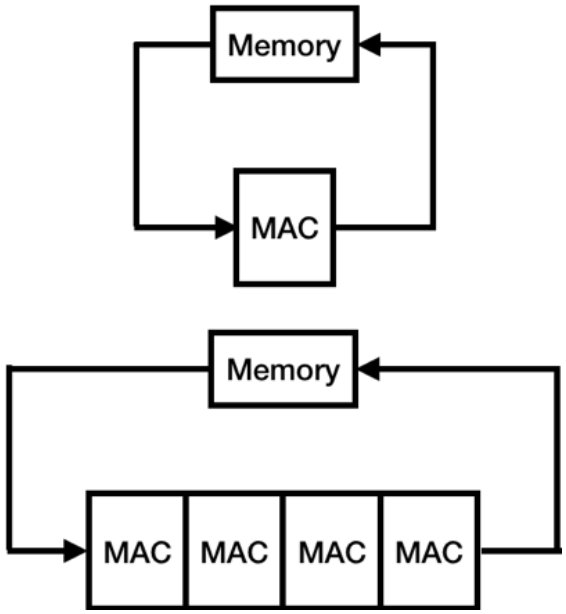

Figure 6: Data flow improvement of the digital MAC array

Chen et al.[8] proposed the CNN chip 'Eyeriss' in 2017 and Jouppi [9] proposed 'TPU' in 2018. Both CNN chips are based on digital circuit, using 2-D MAC array to enhance the speed of the chip and reduce the power consumption. As shown in

Figure 7, ‘Eyeriss’ is using row stationary (RS) data flow on spatial architecture with 168 processing elements to minimizes data movement. ‘TPU’ is using systolic matrix structure to improve the data flow in MAC array.

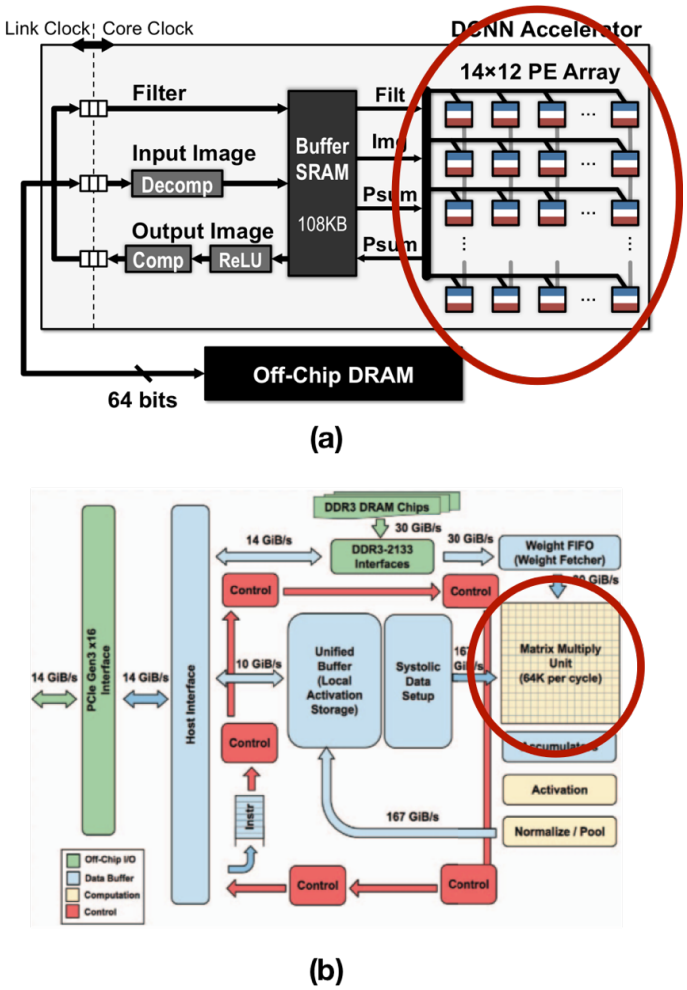

Figure 7: (a) ‘Eyeriss’ top-level structure with PE array[8]; (b) ‘TPU’ top-level structures with ‘Systolic Martix’[9]

On the other hand, the digital structure still requires large area and high power to support the chip working in high precision, which is not a good choice to use on mobile devices or edge applications.

2.2.2 Analog MAC

Analog design is superior to digital in terms of power and area for applications that require ¡8-bit precisions[19]. The art-of-state analog MACs covers almost domain and techniques. Bankman et al.[10] proposed the 8-bit analog MAC based on switched-capacitor, Chatterjee proposed the 3-8 bits current domain MAC [11] and Biswas come up with the voltage domain MAC in 1-bit Weight[12].

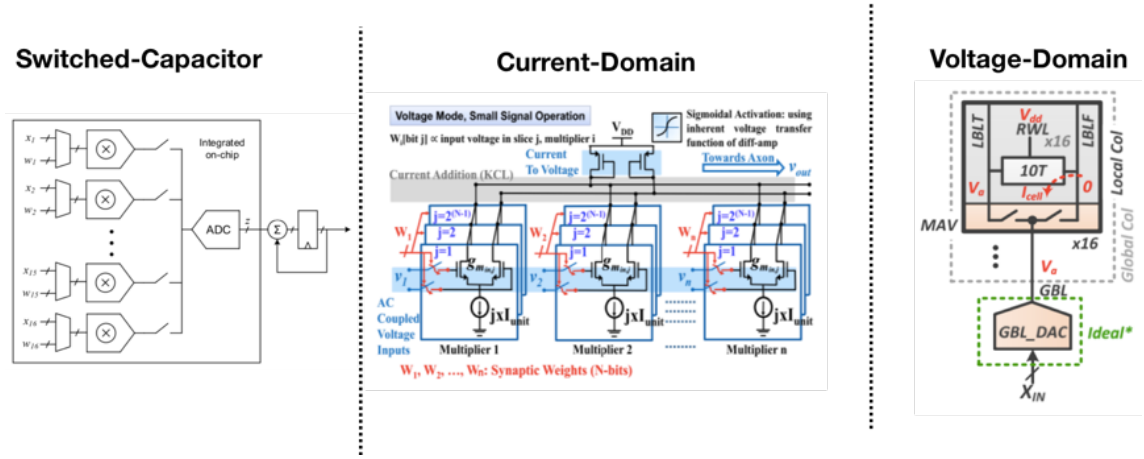

Figure 8: Proposed MAC solutions based on analog circuit [10; 11; 12]

Particularly, in time domain, there is no scale limited of voltage or frequency, which resulting high operation speed with acceptable power consumption. Everson et al. [14] proposed time domain MAC in 3-bit precision by using serial 2X time amplifiers in 2018 as shown in Figure 9. The design is limited by the number of time amplifier, too much time amplifiers means larger power consumption and higher non-linearity from noise.

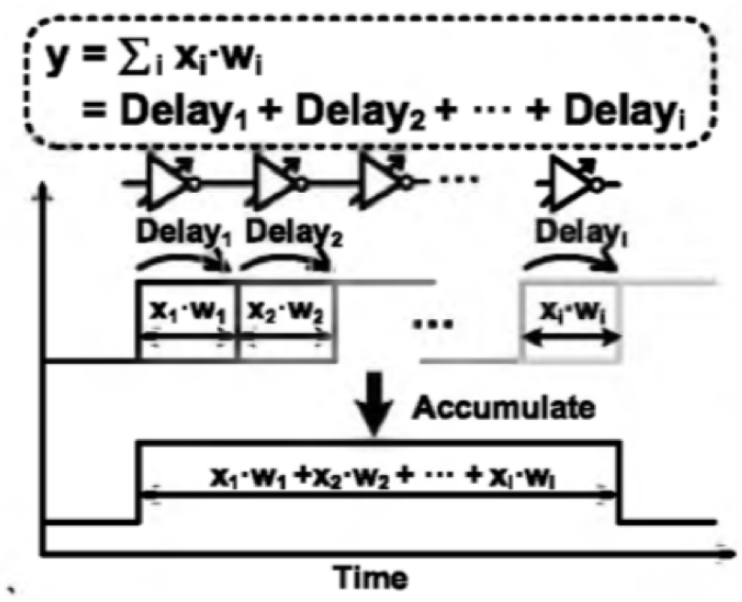

Figure 9: Time Amplifier based MAC[14]

Sayal et al. [13] presented an 8-bit input/1-bit weight MAC with specific memory delay line in 2019 shown in Figure 10. The structure uses AND gate to perform multiply operation which cannot be modified (increase precision).

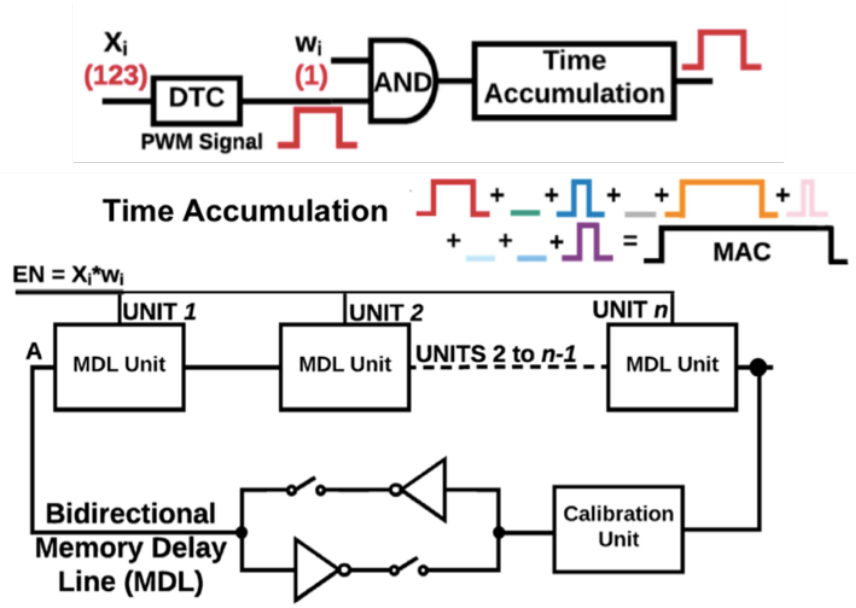

Figure 10: 1-bit weight MAC with memory delay line[13]

Higher precision means better performance and less convolutional layers in CNN models, but also bring higher power consumption and bigger block area. The analog MAC design is finding the balance between precision and circuit requirements.

2.3 Performance Parameter

2.3.1 Integral Nonlinearity (INL) & Differential Nonlinearity (DNL)

The INL is a measure of the deviation between the ideal output value and the actual measured output value for a certain input code. The DNL is a measure of the worst case which deviation

from the ideal 1 LSB step[20].

Both INL and DNL are commonly used to measure the performance of the digital-to-analog and analog-to-digital converters.

2.3.2 Power Efficiency

The power efficiency is an important parameter to describe the efficient of the MAC unit to process one operation, which is calculated by:

$$Power\ Efficient = Total\ Power\ /\ Operations \tag{3}$$

3 Proposed Circuit Design

Based on the literature reviews of the art-of-state designs, this thesis proposed a 6-bit time-domain MAC array.

3.1 Top-level Structure

The proposed MAC array consists of the serial time-domain MACs and two time-to-digital convertors (DTC), connecting by 'positive' and 'negative' lines, as shown in Figure 11. Tr is the reference pulse signal. $Input$ and W are calculated numbers in two matrixes, each of them is 6-bit (1-bit sign, 5-bit value), the sign bit is sent to the line selection control block and the value bits are sent to the time-domain MAC together. The number of time-domain MACs in MAC array is based on the size of the matrix.

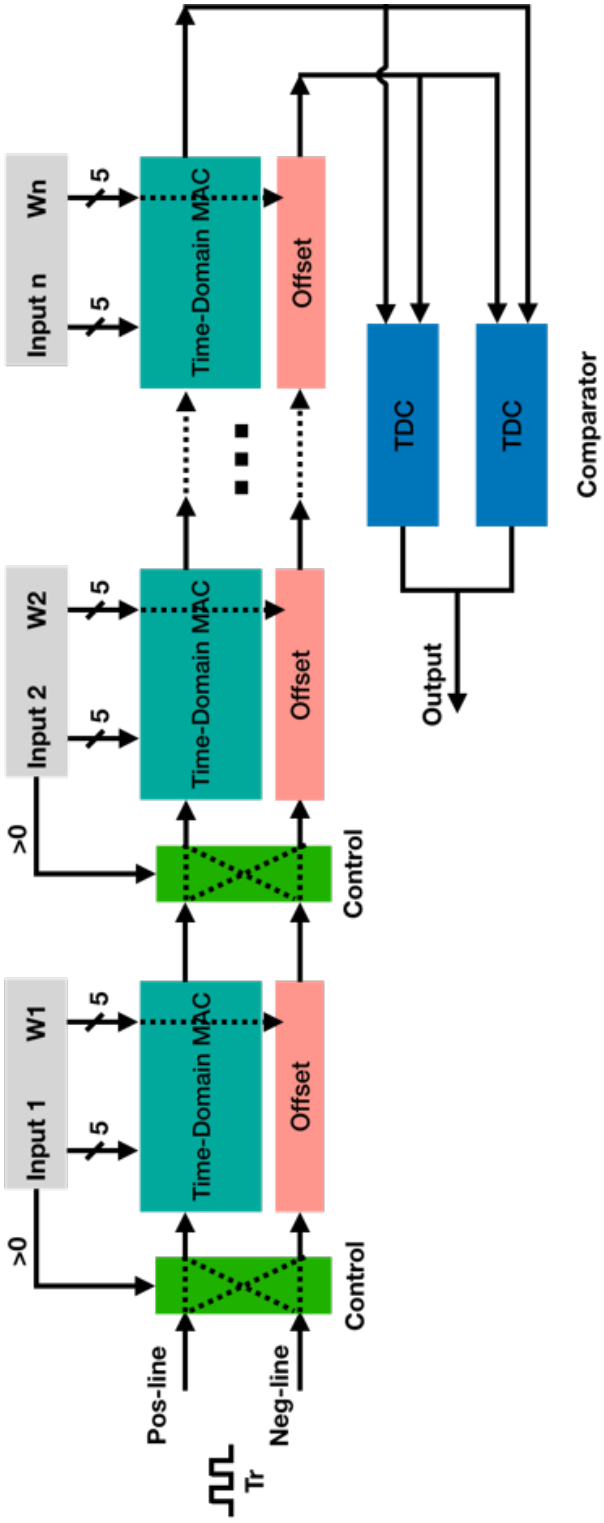

Figure 11: The schematic of the top-level structure

Tr signal is sent to the 'positive' line and the 'negative' line separately at the beginning. The line selection control block chooses a signal on one line and sent it to the next MAC as the input signal, meanwhile, send signal on another line to the next offset block. The selection from the control block is decided by the signs of *input* and W. If two signs are same, the control block will select the signal on the 'positive' line as the input of the next MAC and if two signs are different, the signal on the 'negative' line will be selected.

The selected signal is accumulated a specific time delay according to the $Input$ code and W code in the MAC. At the same time, considering the influence from offset, the same offset of the time-domain MAC is added to the opposite line in offset block. So the total offset delay is same in both lines which can be cancelled in TDC.

The conceptual timing diagram of times delays on different lines is shown in Figure 12, the delay on the 'positive' line is the sum of the positive results (with all offset delay) and the delay on the 'negative' line is the sum of the absolute value of the negative

results (with all offset delay). However, the difference between two delays is uncheckable and TDC can only work when input signal has a positive delay compared with the reference delay. A pair of TDCs is proposed to work together as a comparable-converter, two delay signals is sent to two TDCs in opposite way, the result can only received from one TDC and the sign of the result can be determined by checking which TDC is working. The offset delay is cancelled during the 'subtraction' operation in TDC.

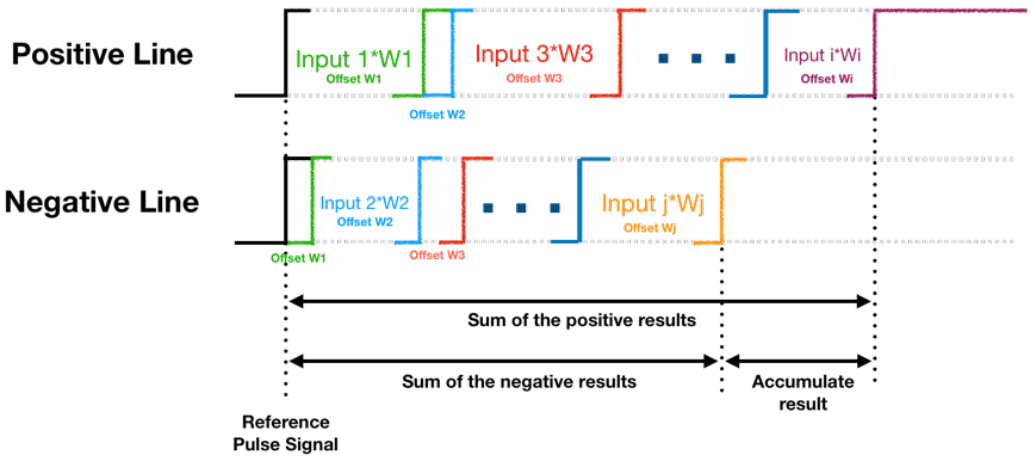

Figure 12: The conceptual timing diagram of how MAC convert digital numbers to time delay and transmit on 'positive' and 'negative' lines

3.2 Time-Domain MAC

The core component of the proposed MAC array is the time-domain MAC because the quality of the single MAC determines the performance of the whole structure. This thesis focus on designing the time-domain MAC based on the constant-slope and variable-slope techniques. The Figure 13 shows the time-domain MAC block, where the input of the MAC is the pulse single from previous MAC, offset block or reference signal, the 5-bit *Input* and *Weight* are control codes of the MAC, which related to a specific time delay.

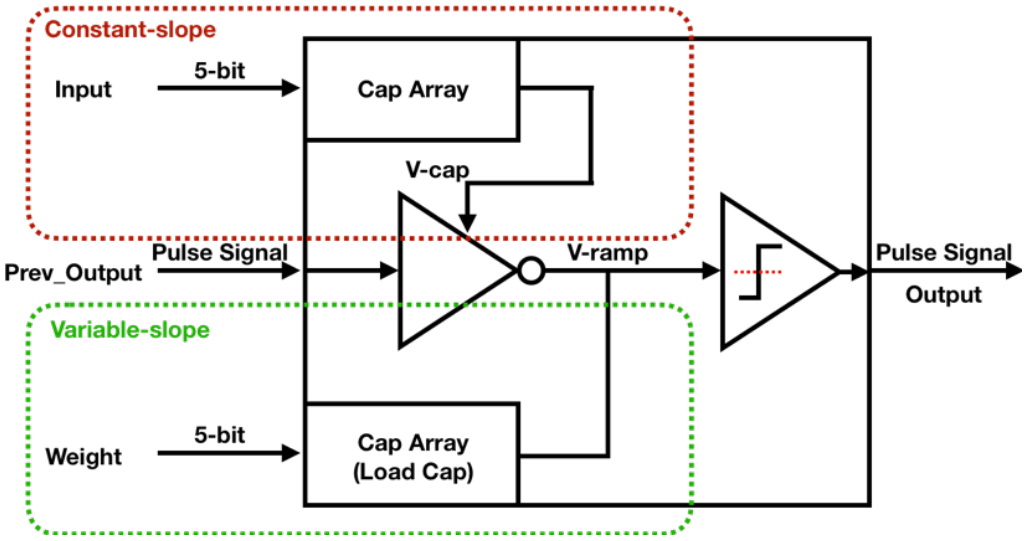

Figure 13: Proposed Time-domain MAC block

The schematic design of the time-domain MAC is shown in Figure 14, which is mainly divided to two parts, based on two different techniques. Both techniques used to design digital-to-time converter (DTC) and achieve high performance with single control code. The proposed time-domain MAC is combining two different techniques to generate an expected time delay controlled by two control codes, processing the multiply and addition operations in time domain.

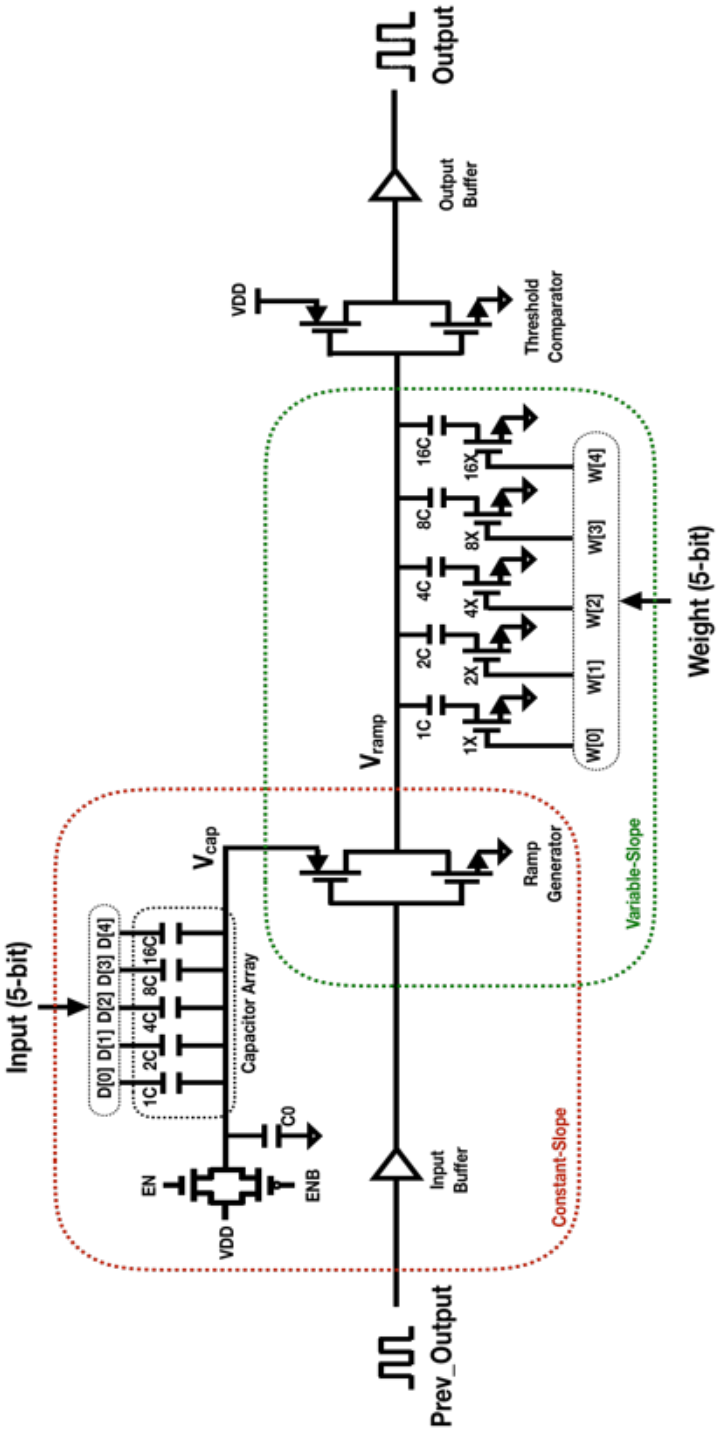

Figure 14: Schematic design of the proposed time-domain MAC

3.3 The Principle of the Design

As shown in Figure 13 and Figure 14, the proposed MAC contains two inverters, the first one is called ramp generator, the second one is called threshold comparator. If the input of the MAC is in rising edge, the ramp generator will generate a signal in falling edge. The constant-slope part controls the supply voltage of the ramp generator which is the start voltage of the falling edge signal and variable-slope part controls the speed of the falling edge, from start voltage to ground. With the different start voltages and falling speeds, different time delays are generated. Then, the generated falling edge signal is sent to the threshold comparator, which inverts the falling edge signal back to the rising edge with time delay.

3.3.1 Constant-Slope Technique

The principle of the constant-slope technique is shown in Figure 15, during the falling edge, different start voltage does not influence the falling slope, but change the time that voltage access the threshold value, generating the different time delay.

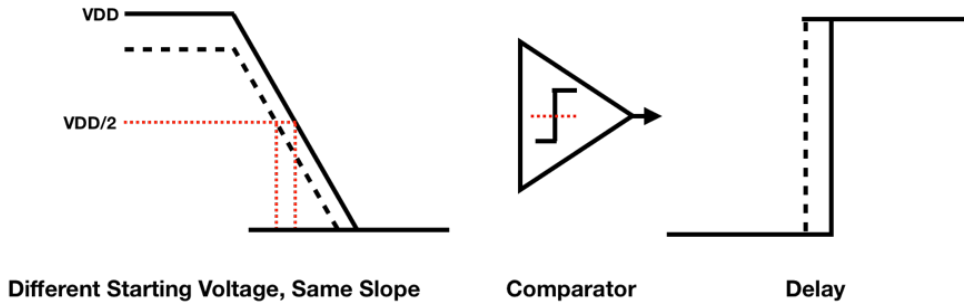

Figure 15: Principle of the constant-slope technique

Inspired by Chen et al. [21], a capacitor array is used to convert the digital code to the voltage which is called capacitor based digital-to-analog convertor (C-DAC). Figure 16 shows how the C-DAC work. At the state 1, capacitors are connected to the VDD (control code is 11111), the power supply (left side, VDD) only charge the capacitor C0, the Vcap is VDD. Then, at the state 2, change the control code (for example: 10101) and cut off the VDD supply, the voltage discharge from C0 and charge capacitors which the code is correspond to 0. Vcap decreases to the desired value, where decreased voltage is decided by the size of the charged capacitors. Before change to the next control code, the C-DAC need to be reset and charge the C0 again at state 3.

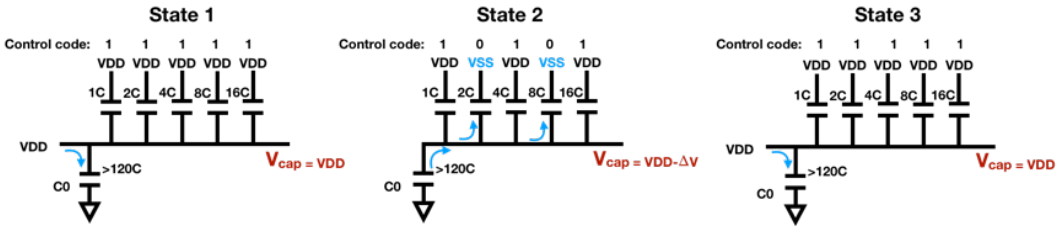

Figure 16: Working states of the constant-slope technique

3.3.2 Variable-Slope Technique

Compared to the constant-slope technique, the variable-slope technique [22] does not change the start voltage of the falling edge but change the falling speed (slope) to generate different time-delay as shown in Figure 17.

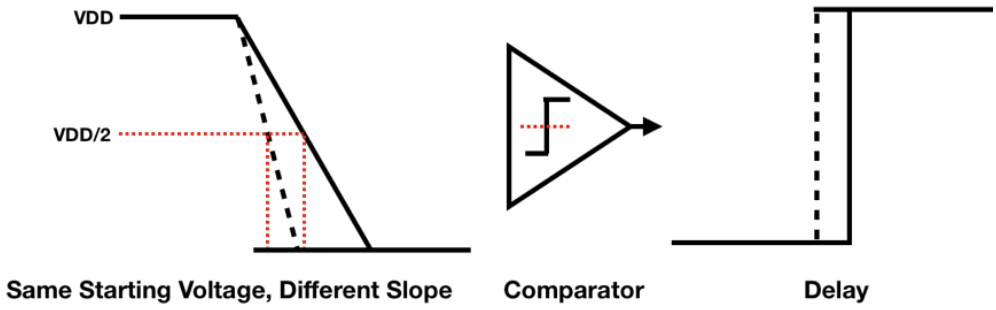

Figure 17: Principle of the variable-slope technique

The principle to change the slope of the falling edge is to append parallel capacitors as the load to the ramp generator. If the input of the ramp generator is 0 (connect with ground),

the output of the ramp generator will be VDD and charge the capacitors which the code is correspond to 1 (NMOS is working and capacitor connect to the ground). When input voltage is changed from VSS to VDD, which is a rising edge, the ramp generator will generate a falling edge signal. At the same time, the capacitors are discharging, which decrease the speed of the falling edge as shown in Figure 18.

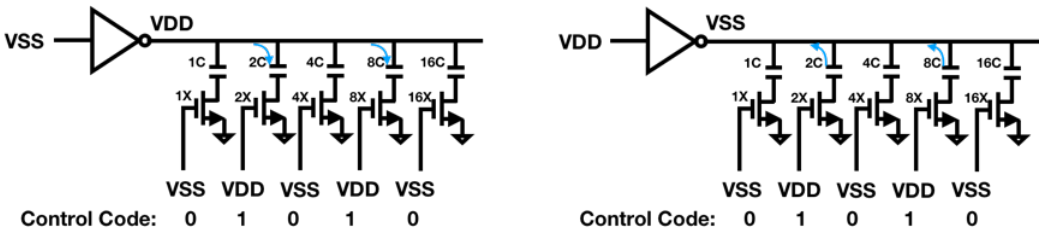

Figure 18: Working states of the Variable-slope technique

3.3.3 The Detailed Operation of the Proposed Technique

The control code of the constant-slope part is the value of the Input, which generate the basic time delay. Then, the basic delay is increased proportionally correspond to the Weight as shown in Figure 19. However, when both $Input$ and $Weight$ are 00000, an offset will be generated caused by the nature of the CMOS inverter. The Weight not only multiply with the basic delay, but also the offset delay.

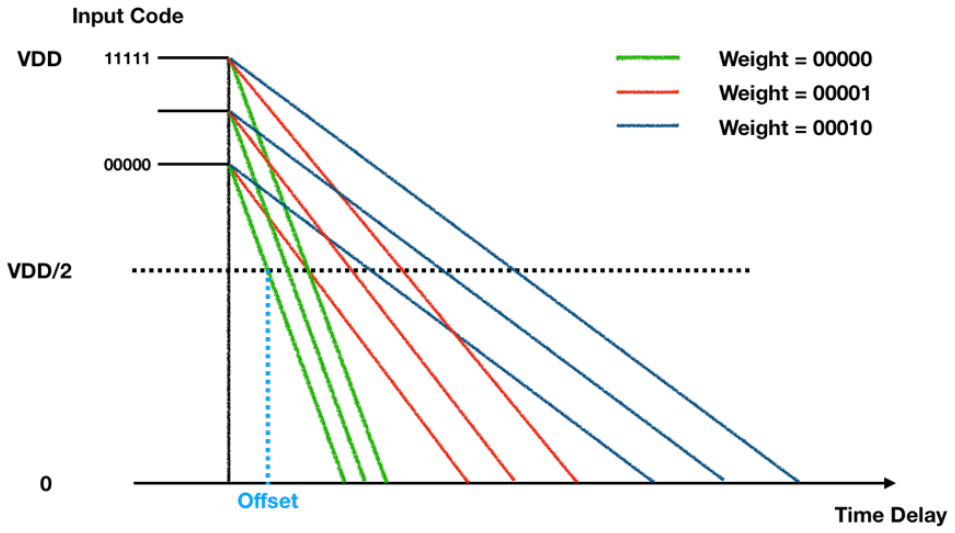

Figure 19: The time delay correspond to the control codes

In the ideal case (without offset), the output is expected as in Figure 20, $Weight$ code is kept same and change the $Input$ code every cycle. The time delay when $Weight$ is 11111 is 31 times as $Weight$ equal to 00001. Consider about the existence of the offset, the trend of the delay will move up in the diagram, however, the multiplication relationship between different delay ranges will not change.

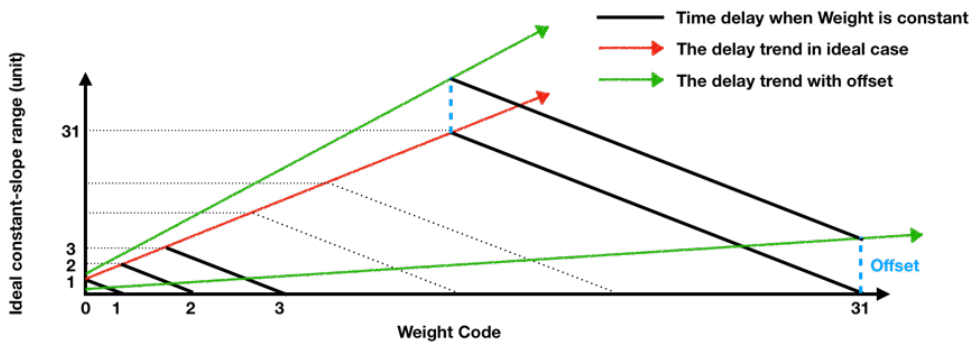

Figure 20: Ideal output & Ideal output with offset

The time schedule also is an important part for combination technique. As shown in Figure 21 (a), at the state 1, the input signal is VSS (low level voltage), EN is connect to VDD (high level voltage), set $Input$ code to 11111 and $Weight$ code to corresponded numbers (e.g. 10101). In this state, the supply voltage (VDD) connects to the ramp generator (invertor) directly as the Source of the PMOS, PMOS is working when of the Gate of the PMOS is connected with input which is VSS. In Figure 21 (a), the red lines mean the voltage on these lines is VDD, blue lines mean the voltage on these lines is VSS. Capacitor C0 and the chosen load capacitors (variable part) are charged. The load capacitors are changed at this state (correspond to the speed of the falling edge).

At state 2, connect EN with VSS, set $Input$ code to corresponded numbers (e.g. 10101) as shown in Figure 21 (b). The supply voltage disconnects with ramp generator, the chosen capacitors in capacitor array is charged from capacitor C0, meanwhile, the Vcap decreases but still be high level because C0 is much larger than sum of the capacitor array. The supply voltage is changed at this state (correspond to the start voltage of the falling edge).

At state 3, input is changed from VSS to VDD, which means a rising edge of the input as shown in Figure 21 (c). The ramp generator is working in a dynamic model and generate a falling edge single with different time delay as mentioned in previous techniques.

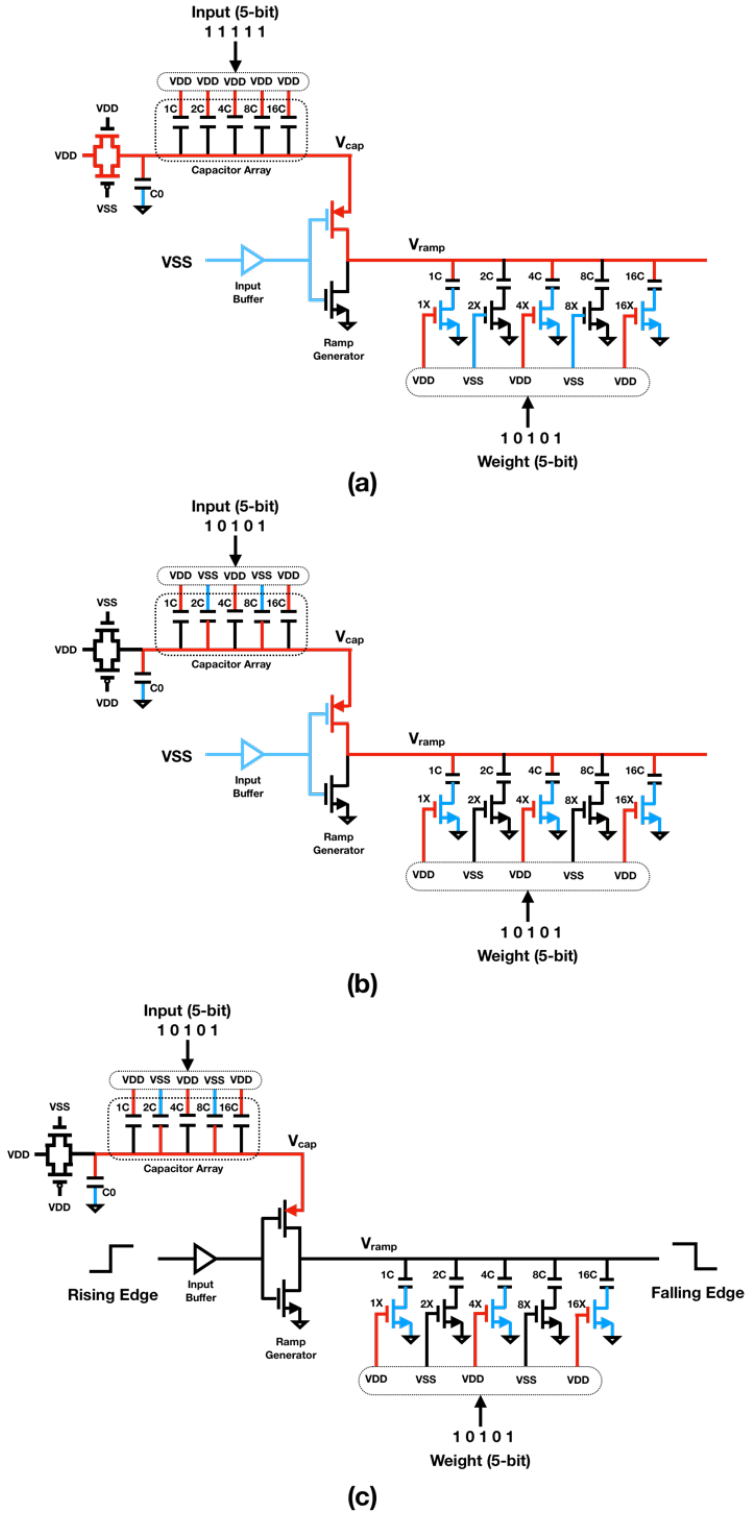

Figure 21: Different states of combination technique

3.4 LSB Requirements

According to the constant-e technique, the LSB can reach to 100 fs in 5-bit precision [16]. To match the multiply operation requirement, the LSB of variable-e technique should larger than 100 * 32 = 3.2 ps, the minimum range of variable-e technique is 3.2 * 32 = 102.4 ps. The precision of the output of the TDC is also 5-bit, considering the matrix size is 3 * 3, which means the resolution of the TDC is 102.4 * 9 / 32 = 28.8 ps in ideal case.

4 Simulation and Results Analysis

Based on the principle of the proposed design, the test bench is built in CADENCE. The block of the MAC has 4 input signals (Vin, EN, Weight¡4:0¿, Input¡4:0¿) and one output. The input signals are supplied by a digital functional block (Verilog), the time diagram is shown in Figure 22.

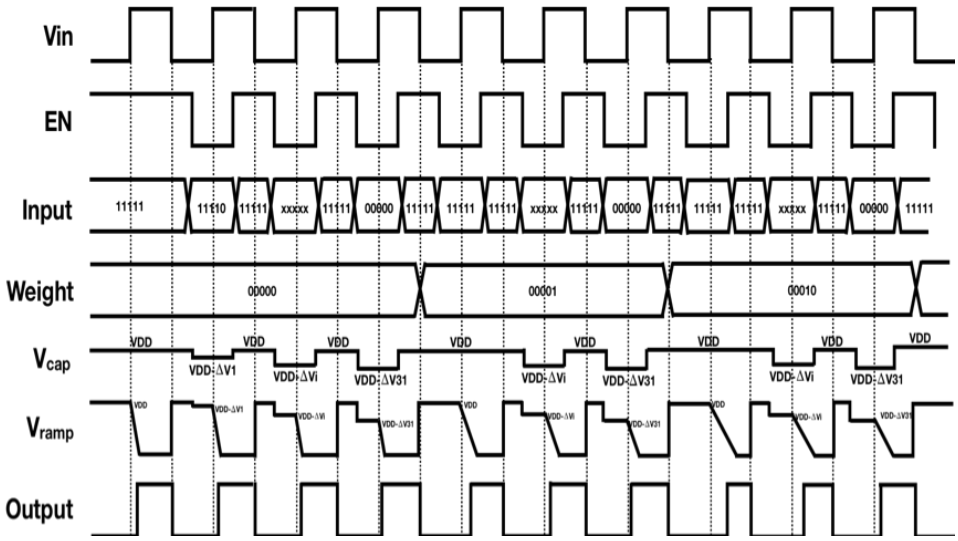

Figure 22: Time diagram of the simulation

According to the time diagram, the simulation scans all possibilities of the combination between *Input* and *Weight*. The *Input* is changed every cycle and *Weight* is changed every 32 cycles (5-bit), the result is shown in Figure 23.

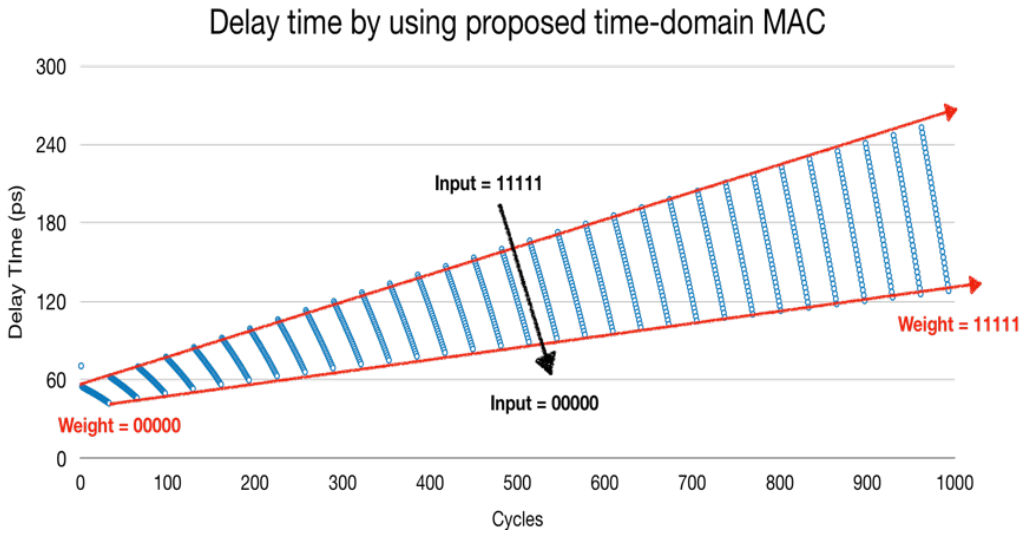

Figure 23: Simulation result

The result shows the good trend, however, the delay range of weight equal to 11111 is 10.1 times of weight equal to 00001 which is smaller than expected magnification value, 31 times. It is because the result range of the constant-slope part is mismatched with variable-slope part.

The performance of each delay slope also be checked. The detail of two slopes are shown following: When $Weight$ equal to 00001:

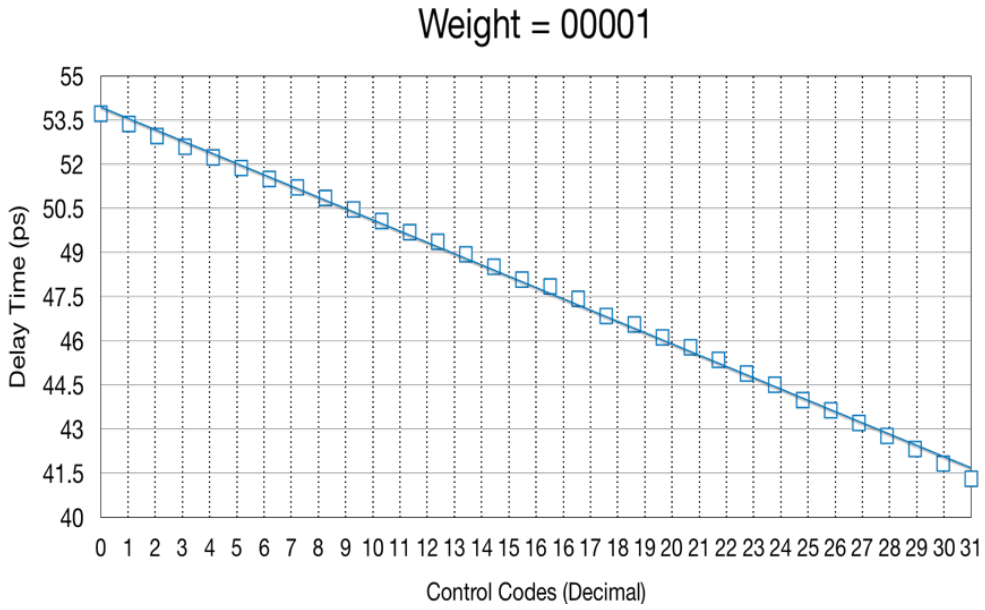

Figure 24: Time Delay when Weight equal to 00001

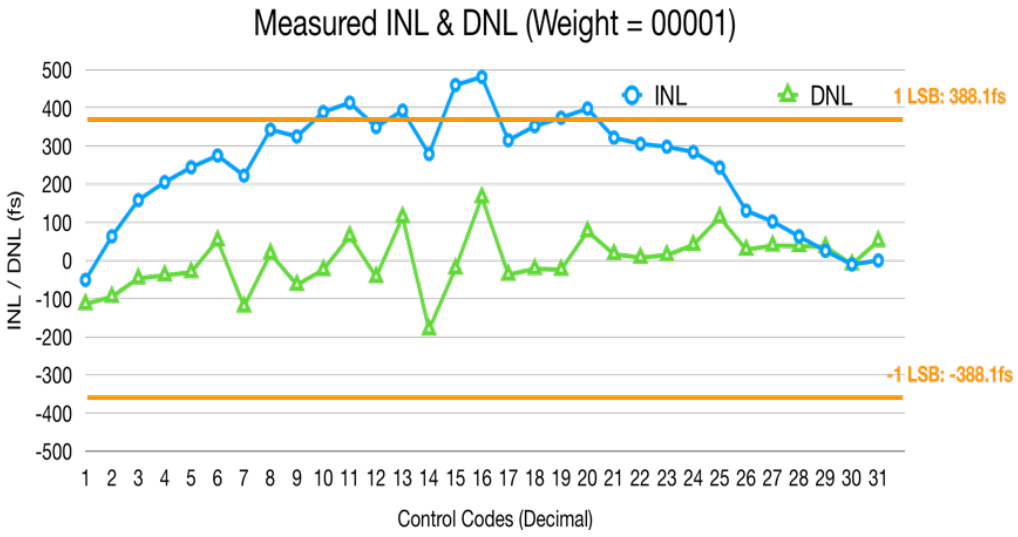

Figure 25: INL& DNL of the time delay when Weight equal to 00001

The time delay range is 12.4 ps and the LSB is 388.1 fs when weight is equal to 00001, which are larger than the work form Chen [21], because the specification of the ramp generator is different and add one load capacitor increased delay time. The peak value of the INL is 1.24 LSB, which is acceptable.

When $Weight$ equal to 11111:

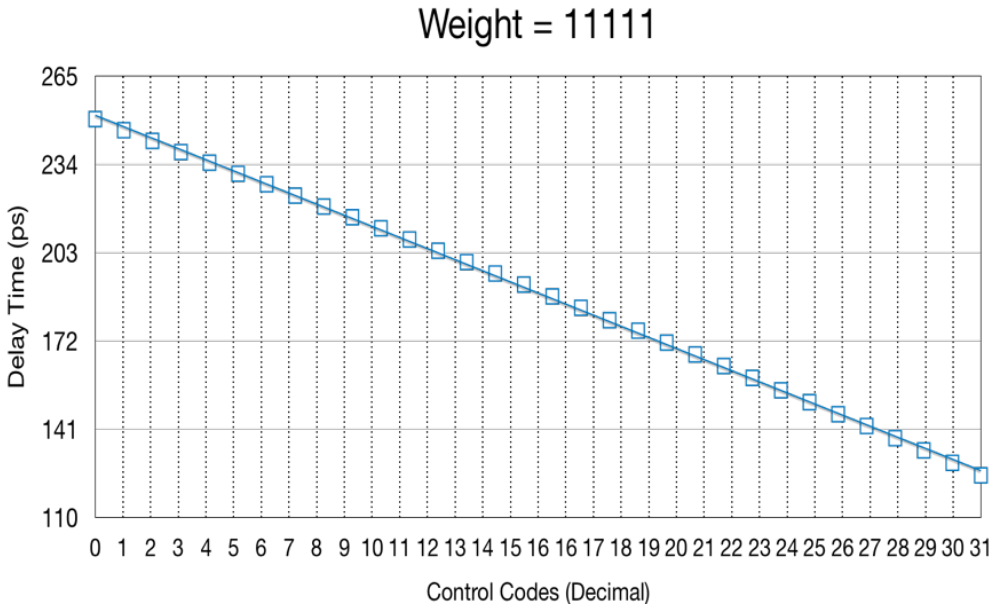

Figure 26: Time Delay when Weight equal to 11111

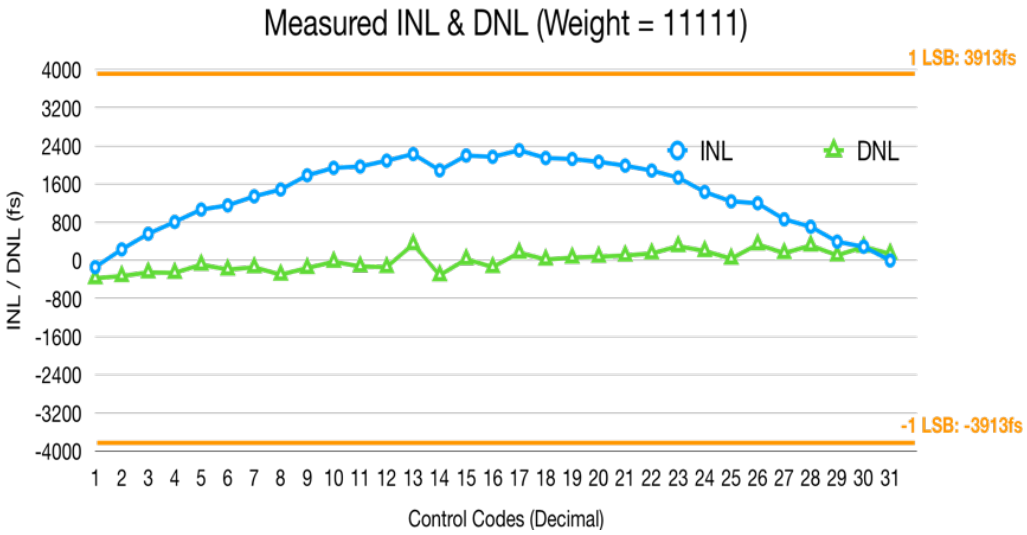

Figure 27: INL& DNL of the time delay when Weight equal to 11111

The time delay range is 125.205 ps and the LSB is 3.913 fs when $Weight$ is equal to 11111, which reach the LSB requirement mentioned in 3.4. The peak value of the INL is 1.24 LSB, which is acceptable. The INL and DNL show the good linearity of the results which are lower than 1 LSB.

5 Discussion and Conclusion

For deploying CNN modules to the edge, the time-domain circuit is a considerable choice to handle the large number of computation operations in CNN models. The proposed 6-bit MAC achieves 0.833 TOPS/W power efficiency at 50 MHz in 28nm CMOS process by combining constant-slope technique and variable-slope technique together. Comparing with other analog MACs, the proposed design achieves high precision and throughout. Moreover, it is a voltage scaling friendly design, also great potential in frequency and precision. Theoretically, the frequency of the design can increase to 1.5GHz (30X) which can significantly improve the throughput of the design.

Table 1: Performance Summary

Technology	28 nm
I/O type	Time
Frequency	50MHz
Supply	1V
Power	120 μW
Efficiency	0.833 TOPS/W
Resolution	5-bit signals & 1-bit sign

However, the delay range of the design does not achieve the expected results causing by the mismatch between constant-slope technique and variable-slope technique, which should be easy to solve by testing different parameter values. The linearity of the results can be improved and the offset need to be cancelled in the future.

6 Suggestions for Future Work

In proposed time domain MAC array, there are three more blocks that need to be design: Lines selection control block, Offset block, TDC. The Lines selection control block can be built with digital logic blocks, as shown in Figure 28.

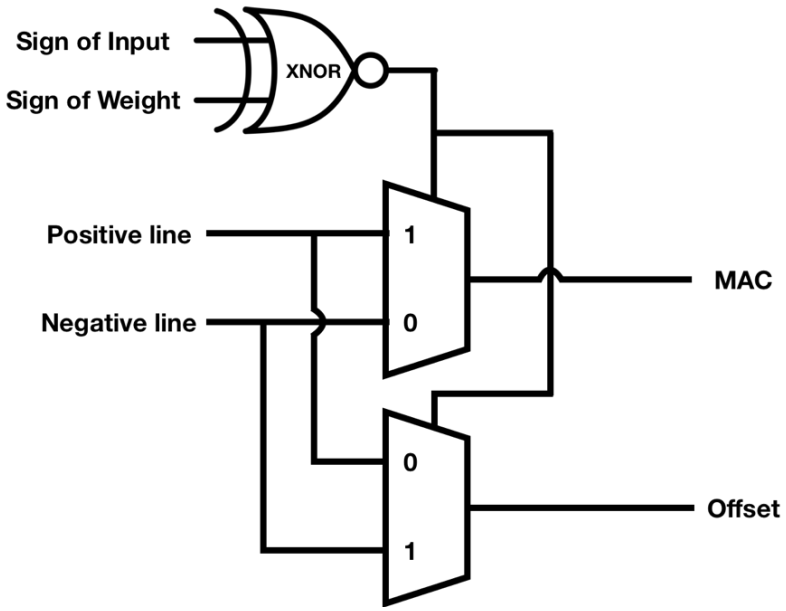

Figure 28: Lines selection control block

The Offset block is similar with the MAC, the constant-slope part can be ignored because the offset delay is only influenced by variable-slope part (multiply operation). The block only needs to be supplied with lowest supply voltage as constant-slope part

can generate (Input code is 00000). The proposed schematic of the offset block is shown in Figure 29.

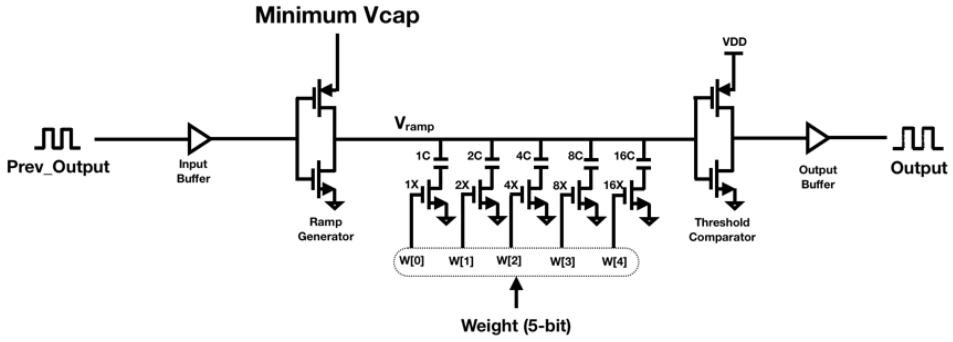

Figure 29: Proposed schematic of the offset block

The TDC structure is inspired by Minas [23] shown in Figure 30. A start signal is fed into the delay line first and then a stop signal is sent to sample the state of the delay line. By the time the stop is sent, the start already propagates to some point of the delay line, making the first half of the delay elements in high voltage and the other half in low voltage. The precision of the TDC is kept in 5-bit, which fits the normalization step in CNN models.

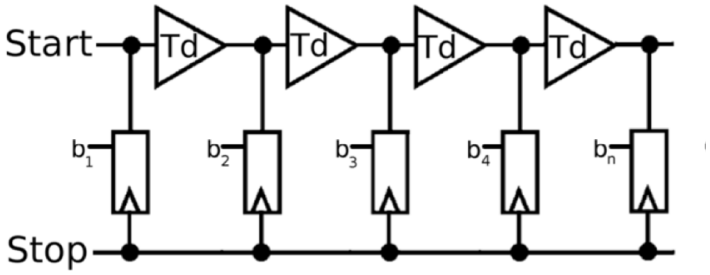

Figure 30: TDC art-of-state architecture[23]

The MAC array is just the first step to build the complete CNN accelerating chip. The calibration circuit will be built to reduce the INL & DNL of the results. The frequency could be increased to Gage Hz level. The top-level structure can be built based on the proposed solution to test with the real CNN tasks (digital number recognition, image classification). According to the digital CNN chips architecture, the data flow improvement also can be used in analog chips to reduce the reading and writing energy between MACs and memory. Moreover, the analog chips have potential to work with sensors directly before digitizing the input data.

References

[1] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[2] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[3] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "{GAZELLE}: A low latency framework for secure neural network inference," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1651–1669.

[4] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.

[5] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving

the speed of neural networks on cpus," in *Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*, 2011.

[6] M. Courbariaux, I. Hubara, D. Soudry, R. El-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.

[7] R. Pawar and D. Shriramwar, "Review on multiply-accumulate unit," *International Journal of Engineering Research and Applications*, vol. 7, no. 06, pp. 09–13, 2017.

[8] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2016.

[9] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *2017 ACM/IEEE 44th Annual International Sym-*

posium on Computer Architecture (ISCA). IEEE, 2017, pp. 1–12.

[10] D. Bankman and B. Murmann, "An 8-bit, 16 input, 3.2 pj/op switched-capacitor dot product circuit in 28-nm fdsoi cmos," in *2016 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE, 2016, pp. 21–24.

[11] B. Chatterjee, P. Panda, S. Maity, A. Biswas, K. Roy, and S. Sen, "Exploiting inherent error resiliency of deep neural networks to achieve extreme energy efficiency through mixed-signal neurons," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 6, pp. 1365–1377, 2019.

[12] A. Biswas and A. P. Chandrakasan, "Conv-ram: An energy-efficient sram with embedded convolution computation for low-power cnn-based machine learning applications," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2018, pp. 488–490.

[13] A. Sayal, S. Fathima, S. T. Nibhanupudi, and J. P. Kulkarni, "14.4 all-digital time-domain cnn engine using bidirec-

tional memory delay lines for energy-efficient edge computing," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 228–230.

[14] L. R. Everson, M. Liu, N. Pande, and C. H. Kim, "A 104.8 tops/w one-shot time-based neuromorphic chip employing dynamic threshold error correction in 65nm," in *2018 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE, 2018, pp. 273–276.

[15] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 253–256.

[16] M. ul Hassan, "Vgg16-convolutional network for classification and detection," 2018. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge,"

International journal of computer vision, vol. 115, no. 3, pp. 211–252, 2015.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] R. Sarpeshkar, "Analog versus digital: extrapolating from electronics to neurobiology," *Neural computation*, vol. 10, no. 7, pp. 1601–1638, 1998.

[20] W. M. Sansen, *Analog design essentials*. Springer Science & Business Media, 2007, vol. 859.

[21] P. Chen, F. Zhang, Z. Zong, H. Zheng, T. Siriburanon, and R. B. Staszewski, "A 15-μw, 103-fs step, 5-bit capacitor-dac-based constant-slope digital-to-time converter in 28nm cmos," in *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE, 2017, pp. 93–96.

[22] N. Markulic, K. Raczkowski, P. Wambacq, and J. Craninckx, "A 10-bit, 550-fs step digital-to-time converter in 28nm cmos," in *ESSCIRC 2014-40th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 2014, pp. 79–82.

[23] N. Minas, D. Kinniment, K. Heron, and G. Russell, "A high resolution flash time-to-digital converter taking into account process variability," in *13th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'07)*. IEEE, 2007, pp. 163–174.

APPENDIX

A. Code of the digital control (Verilog)

```
1   `timescale 1ps/1fs
2
3   module DTC_ctl(EN,Vin,Input,Weight);
4   output reg EN;
5   output reg Vin;
6   output reg [4:0] Input;
7   output reg [4:0] Weight;
8
9   reg [4:0] trans;
10  reg clk;
11  parameter T=20000;//20ns, 50MHz
12
13  initial begin
14  clk <= 1;
15  Vin <= 0;
16  EN <= 1;
17  Input <= 5'b11111;
18  Weight <= 5'b00000;
19  trans <= 5'b11111;
20  end
21
22  always #(T/2) Vin <=~ Vin;
23
24  always #(320000) clk <= ~clk;
25
26  always @(posedge clk) Weight <= #(12500) Weight + 1'b1;
27
28  always @(posedge Vin) trans <= trans - 1'b1;
29
30  always @(posedge Vin) begin
31          Input <= #(4000) 5'b11111;
32          Input <= #(16000) trans;
33  end
34
35  always @(posedge Vin) begin
36      EN <= #(5000) 1;
37      EN <= #(15000) 0;
38  end
39
40  endmodule
```

Figure 31: The code of the digital control

B. Library and Testbench Location

Server	icecream2
User	xwu
Library	/home/xwu/Design_xwu/Design/time_domain_MAC
Testbench Cell	DTC_v3_TB