2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC) | 978-1-6654-9604-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMACC54824.2022.10093588

# Efficient Re-configurable Multiply and Accumulate Unit for Convolutional Neural Network

Pavankalyan Chintapoodi
*Electronics and Communication Engineering*
*Koneru Lakshmaiah Education Foundation*
Vaddeswaram, India
180040645ece@gmail.com

Ramakrishna Thirumuru
*Electronics and Communication Engineering*
*Koneru Lakshmaiah Education Foundation*
Vaddeswaram, India
ramakrishnathirumuru@gmail.com

Muzammil Parvez M
*Electronics and Communication Engineering*
*Koneru Lakshmaiah Education Foundation*
Vaddeswaram, India
parvez190687@yahoo.co.in

Ali Baig Mohammad
*Electronics and Communication Engineering*
*REVA UNIVERSITY*
Bengaluru, India
mdabaig@gmail.com

G. Sudhagar
Electronics and Communication Engineering
Bharath Institute of Higher Education and Research
Chennai, India sudhagar.ece@bharathuniv.ac.in

***Abstract*—Conventional Multiply and Accumulate (MAC) unit consists of multiplier, and accumulator. The hardware realization on programmable system on chip of the systems require trade-off between speed and area. In this paper, the Booth Multiplier, array multipliers, ripple-carry array multipliers are expiored with row bypassing technique, Vedic multiplier, Wallace-Tree multipliers, and DADDA multipliers in terms of area, delay, and power. The combination of Radix-4 Booth multiplier and carry-s adder has shown better performance in terms of area, and delay in the design of the MAC Unit. The design is then synthesized on ARTIX-7 FPGA using the Xilinx Vivado. Further, the MAC unit has been evaluated in implementing a 2D convolution engine application.**

***Index Terms*—Re-configurable hardware, ARTIX FPGA, MAC unit, Convolution engine**

## I. Introduction

In the domains of AI, machine learning, and deep learning, neural networks resemble the function of the human brain, allowing computer programs to spot patterns and solve common problems. In recent times, Digital Signal Processor (DSP) architectures are being implemented on reconfigurable hardware for faster operations. Now, the fundamental consideration of MAC design has been to increase speed of the neural networks. As a result, a high-throughput MAC Unit is always an essential component for obtaining high performance inreal-time signal processing applications. Convolutional Neural Networks (CNN) require a lot of processing and weight data, which puts a strain on the battery.In mobile systems, and researchers have proposed hardware accelerators for CNNs that include numerous parallel hardware Units should be multiplied and accumulated. [1]. Low-power designs are becoming increasingly important as the number of portable electronic items grows. This is due to the restricted battery energy of these portable products, which limits the system's power use. CNN training entails gradually changing the numeric "weight" values associated with the neural network's connections. After training, the weights are not updated again, and the trained network can be deployed to a great amount of embedded devices.Han et al. [2], [3] discovered that in trained CNNs, identical weight values can occur several times. They postulated that binning the weights and retraining the network using the binned values was adequate in many circumstances. They encode the weights by substituting a four-bit index for the original numeric values, indicating which of the 16 shared weights should be utilised. The size of the weight matrices is considerably reduced as a result. The primary goal is to look at different MAC architectures and design methodologies that can be used to implement high-throughput signal processing units. DSP processors are meant to speed up the execution of DSP algorithms by lowering the number of clock cycles required [4]. An examination of these algorithms—for example, in digital filters, data correlation, and computing of the Fast Fourier Transform—shows that optimizing the multiply-and-accumulate (MAC) Units operations is key to achieving good performance, i.e., to reduce the required clock cycles and power consumption. The multi-channel convolution is the most computationally expensive part of a CNN. The administrator's feedback is a multi-channel image, which is joined with an enormous number of parts to frame a result picture with different channels. The deepest iterations can be executed with an MAC equipment, and CNN accelerators frequently have countless concurrent MACs [5]. Melody Nien Tang showed that a gather MAC units for convolutional Neural Networks (CNN) induction tasks that help a double mode truncation error compensation(TEC) strategy in view of a fixed-width Booth multiplier (FWBM). CNNs are a sort of profound learning model that has been exhibited to be successful in an assortment of uses, including image and signal processing, pattern recognition, and computer-vision [6].

Numerous CNN interference tasks are anticipated to be executed in client-side gadgets as edge artificial-intelligence applications,Subsequently, a ton of work has gone into equipment (HW) speed increase moves toward that utilization application- explicit coordinated circuits (ASICs) or field-programmable door exhibit (FPGA) units to execute productive client-side CNN induction calculations [7]- [9]. The fundamental cycles of the convolution and completely associated layers in a CNN [10] are two-dimensional (2D) convolution and inner product calculations, which are led through a progression of duplicate collect tasks (MAC) tasks. Macintosh units are expected to construct the processing elements (PEs) for CNN speed increase [10]- [11]. In view of the CNN order exactness [11]a reasonable bit width might be determined by thinking about the fixed-point MAC activity. One procedure is to enact the duplication yield in a CNN accelerator with a similar piece width as the contribution to stay away from significant bit- width expands because of gathering in a MAC unit [12].

For CNN's , Booth multipliers are typically employed as MAC multiplication algorithms [12]– [13]. According to Tung Thanh Hoang to increase MAC performance, the critical path delay can be reduced by inserting an extra pipeline register, either inside the partial-product (PP) unit or between the PP unit and the final adder. Because carry propagation is time expensive, doing two distinct carry propagation in the same MAC circuit is wasteful. The use of a typical accumulating adder is eliminated by feeding the multiplier output back to the input of the PP unit reduction tree . In general, the first stage of a two-cycle MAC design is much slower than the second (accumulation) stage. A new two-cycle MAC design is prposed with a slower second stage but a much quicker first stage, resulting in a better delay balance between the two stages. The implementation of product sign extension is the key to the new architecture: the sign-extension circuitry, together with the accumulating adder and the saturation unit, is positioned in the second stage. As a result, the product's feedback is contained within the pipeline's second step. In recent years, deep learning [13] has seen a lot of success. Deep learning can attain performance that is close to, if not greater than, that of humans in many applications. Although deep learning is powerful, the expense of implementing one is high [14], especially in terms of computing intensity]. Many studies have been conducted in recent years on the effective implementation of deep learning algorithms on hardware. The numerical format required by deep learning training and inference is widely researched in these publications. Many recent research papers [15] have focused on minimising the numerical precision required by deep neural network training in order to reduce hardware costs.

Zhiming Zhang, Laurent Njilla, explained that the Field Programmable Gate Arrays (FPGAs) paved way for a rapid growth era due to their attractive flexibility and CMOS-compatible fabrication process.According to Global Industry Insights, the FPGA market is estimated to reach 9.98 billion US dollars by 2022 [15]. The increasing popularity of FPGA may drive more attackers to compromise. FPGA-based systems through various channels. The supply chain for modern FPGAs is becoming more worldwide because of efficiency and cost savings. This trend could make it more likely that FPGA devices or FPGA design tools aren't reliable. Theft and modification of intellectual property (IP) can occur in a variety of data forms, including hardware description language (HDL) and bitstream [16]. There are only a few works that address CAD tool assaults on FPGAs. Hardware Trojans implanted by malicious FPGA design suites have been detected using logic testing and side-channel analysis [17]. Multiple Excitation of Rare Occurrence (MERO) is a simple way to produce test patterns for Trojan identification. To detect the hardware Trojan on FPGAs, the research takes advantage of the relationship between dynamic current and maximum operational frequency.

The remaining portion of the essay is organised as follows: The MAC unit's appearance and activity are managed by Section I. The designs of the Convolution core are discussed in Section II. In Section III, the current MAC unit model is explained. In Section IV, various suggested MAC unit models are discussed. Segment V also covers the generation and simulation of MAC unit models. Section VI shows the conclusion and upcoming work.

## II. Proposed MAC unit design

A multiplier has three total functioning steps. Using the multiplicand X and multiplier Y, booth encoding is used to create partial products as the first step. The next stage is partial product compression, which involves adding the partial products into sum and carry bits. The final summing, which takes place in the third stage, involves adding the sum and carry to obtain the final augmentation result. The sum in MAC is then increased by the outcome.

Mathematically, a MAC is expressed as

$$P = (AB) + Z \tag{1}$$

where multiplicand, multiplier, and preceding multiplication result are A, B, and Z, respectively. Based on the equation developed, the suggested MAC design has a hybrid CSA tree. The inclusion of incomplete products is clearly blended with the accumulation stage. The booth encoder converts the n-bits A and B into (n+1) bits partial products.

The MAC design is updated by replacing the final adder with a suggested Square Root Carry Select Adder (SQRT CSLA) based on common boolean logic and employing radix-4 MBA for partial product generation. This is proposed to reduce the MAC's delay even more. When opposed to a conventional booth multiplier, the major goal of utilising a booth encoder is to reduce the number of partial products.By recoding the numbers to be multiplied, booth multiplication reduces the size and speed of the multiplication circuits. By employing the radix-4 Booth recoding technique, the number of incomplete products is reduced by half. Instead of shifting and adding for each column of the multiplier term and multiplying by 1 or 0, and chosen the second column and multiply by 1, 2, or 0. This produces the same results. As a result, the number of partial items is cut in half. By

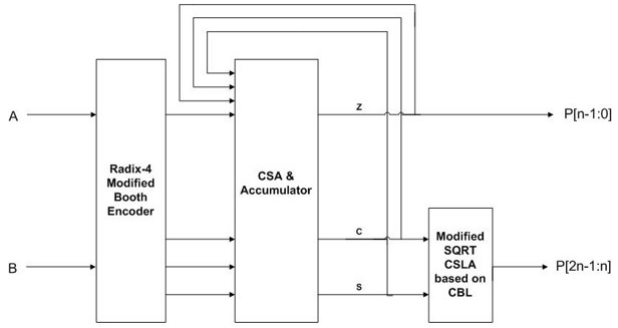


Fig. 1. modified booth multiplier

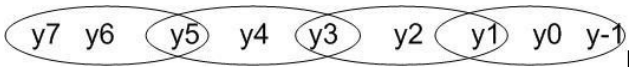


Fig. 2. Grouping of multiplier bits

organising the bits into triplets of three bits, or blocks, and overlapping each block by one bit, the multiplier term is then recoded. The starting block only needs two bits of the multiplier, and the LSB is utilised to begin grouping. Due to its LSB nature, the first block only requires two bits of the multiplier. Figure 2 displays the triplet-shaped collection of bits from the multiplier. The multiplier's encoded digit causes the multiplicand X to undergo a particular operation. The approach is applied separately to each of the two-bit groups that make up the multiplier B. Fig. 3 displays the relevant logic circuit in question. The Booth decoder creates partial products from encoded signals, as seen in Fig. 4.

Adders are a common digital component in the design of digital integrated circuits. In general, addition is the fundamental process found in practically all calculation and computer systems. As a result, the efficient implementation and design of arithmetic units necessitates the efficient implementation of binary adder structures. A ripple carry adder has a smaller design area and slower speed. Because of its large area needs, a carry look ahead adder is faster to operate. Carry a few select adders in the middle of the spectrum. The use of the Square-root technique to design a highly efficient Carry Select Adder reveals various possibilities for boosting the speed and minimising the size of any data processor. In general, the quickest adder is the carry select adder (CSLA), which is utilised in many data-processing processors to accomplish fast arithmetic operations. If the structure of CSLA is examined, it can seen that there is room for improvement in terms of area reduction and delay. A carry choose adder for the computing process is described in this paper, along with various modules that must be developed and synthesised using HDL coding. In comparison to all other adders, the carry select adder (CSLA) is one of the fastest. To lower the area and delay of the CSLA, this review experiences a very simple and efficient gate-level adjustment. In comparison to the normal SQRT CSLA design, 8-bit, 16-bit, 32-bit, and 64-bit Square-Root CSLA (SQRT CSLA) architectures have been constructed based on this alteration. In comparison to the traditional SQRT CSLA, the proposed circuit design has less area and delay.

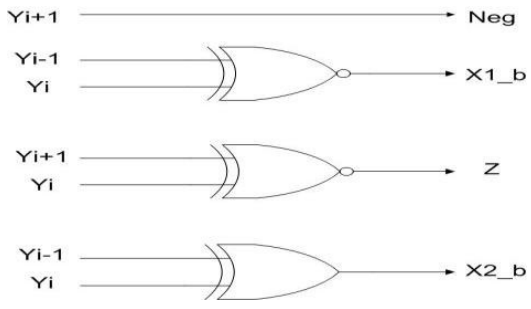


Fig. 3. Booth Encoder logic

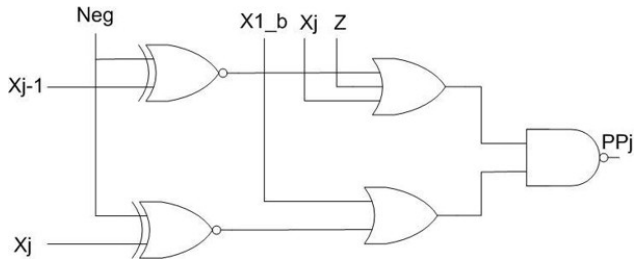


Fig. 4. Booth Decoder logic

The proposed architecture and relevant implementation methodologies have been discussed in this Section. In the MAC unit the Booth multiplier is used. Booth's multiplication algorithm is a two's complement multiplication algorithm that multiplies two signed binary values. Andrew Donald Booth devised the algorithm in 1950 while working on crystallographic research at Birkbeck College in Bloomsbury, London. When compared to regular multiplication, the Booth multiplier finds the operand that works as a multiplier and can execute multiplication for the method since it reduces the number of steps while completing addition. Proposed design implements a fully parameterized convolution operation. By 'fully parameterized',mean to say that the same design can be used to synthesize a convolution layer of any size the architecture demands(conv3x3, conv7x7,conv11x11). In the code just vary the parameters passed to this code before the synthesis.

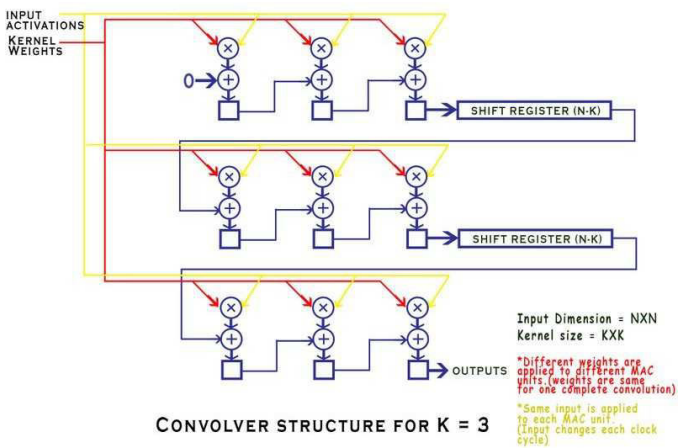


Fig. 5. Convolution Engine

The fundamental goal of this design is to provide a highly pipelined streaming architecture in which the processing module does not have to halt at any stage. i.e., any part of the convolver does not wait for the result of another part's work. Every clock cycle, each stage performs a small percentage of the complete job on a separate section of the input. This isn't simply a feature of this design; it's a general idea known as pipelining which is often used to break down complex computational processes into smaller parts while also increasing the circuit's maximum frequency.

The process of gathering instructions from the CPU is pipelining. It is a way of executing several instructions at the same time that allows for the storage and execution of instructions in an orderly manner. The pipeline is divided into stages, each of which is joined to the next to form a pipe-like structure. Instructions come in from one end and leave from the other.Pipelining improves the total flow of instructions. Each segment of a pipeline system is made up of an input register followed by a combinational circuit. The register is used to store data, and the combinational circuit manipulates it. The output of the combinational circuit is applied to the next segment's input register. The convolver is nothing more than a collection of MAC units and a few shift registers that, when fed the correct inputs, output the convolution result after a certain number of clock cycles.

The input Feature map has a dimension of NxN and that the proposed kernel (filter/window) has a dimension of kxk. It should go with saying that the dimension of the kernal is less than dimension of the image. The stride with which the window advances over the feature map is represented by s. By understanding how 2-D convolution works, the output feature map is less than the input feature map. The input feature map does not contain any zero paddings, the output will be of the size (N-K+1)/s x (N-K+1)/s. As indicated in the diagram, the proposed convolver contains MAC units the size of the convolution kernel and a handful of shift registers coupled at various points in the pipeline. It's important to note that the inputs to these MAC units are different. The parameters are choosen as k=3 and n=4 with s=1 in this work. This methodology is shown in Fig. 1.

The kernel weights are placed in such a way that each MAC unit has a constant weight value throughout the convolution process, while each mac unit has a varied weight value depending on its position.In contrast, the same value is assigned to every MAC unit for input activation values, except that the value varies with each clock cycle according to the input feature map.The proposed design also has a few more signals, such as valid convolution and end convolution, that notify the outside world whether or not the outputs of this convolver module are valid.

## III. Experimental Setup

FPGA, which solves the cost problems associated with ASIC manufacture. Using an off-the-shelf component made up of basic programmable logic parts, FPGAs enable the designer to develop a custom circuit implementation of an algorithm. This platform provides the power savings and performance benefits of lower manufacturing nodes without the expense and difficulty of developing an ASIC as demonstrated in Fig. 3. In a cost-optimized FPGA, Artix-7 devices offer the greatest performance-per-watt fabric, transceiver line rates, DSP processing, and AMS integration. The series provides the best value for a variety of cost and power-sensitive applications, such as software-defined radio, machine vision cameras. As shown in Fig. 2 Artix-7 (XC7A35T) Family with Package CPG236 consist of HR I/O bank 15 is not bonded out, 16, 34, and 35 are partially bonded out, The GTP Quad 216 is partially bonded out.

## IV. Results and Discussion

With reference to the convolution engine demonstrated in Fig. 5 , the convolution outputs are the values that correspond to the state where the validconv signal is high. Valid convolution and end convolution are two signals that indicate the outside world whether the outputs of this convolver module are valid or not.When the window finishes traveling over one row of the input, it wraps around to the next row, resulting in inaccurate outputs.This could be skipped while calculating during the wrap-around phase, but it would need to stop the pipeline, which is against this streaming design approach. So, with the help of a legitimate convolution signal, the outputs are discarded. The schematic for synthesis of the convolution engine and the corresponding simulation results are shown in Fig. 6 and Fig. 7 respectively.

## V. Conclusion

A re-configurable MAC unit has been implemented on XILINX make ARTIX-7 FGPA for the realization of convolution neural network using RTL based design. The coding has been carried out in Verilog, as it allows the description at different

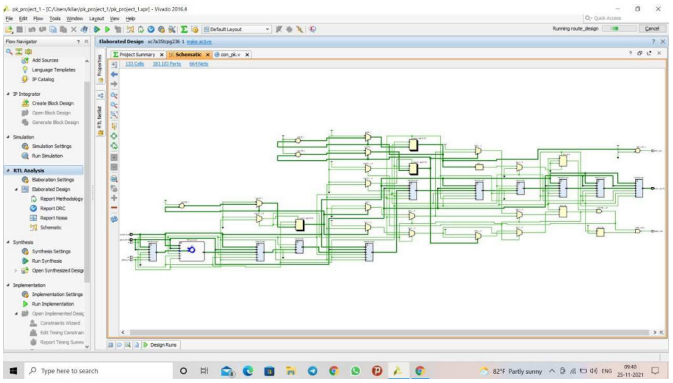

Fig. 6. Schematic of Convolution Engine

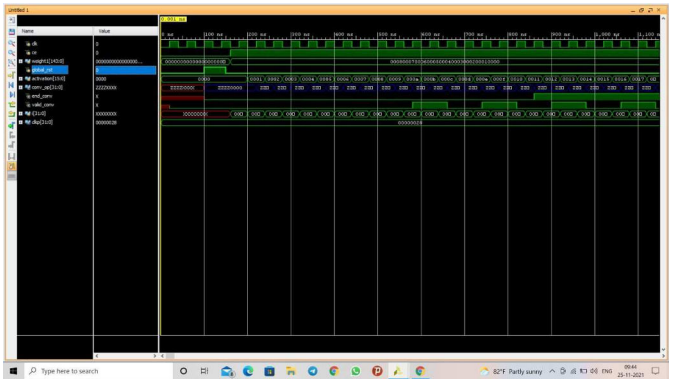

Fig. 7. Simulation results of convolution Engine

levels of abstraction. The functioning of the proposed MAC unit mainly relies on the efficiency of Vedic multiplier and carry-save adder and it has shown better performance in terms of area, and delay.

## REFERENCES


[1] Garland, James, and David Gregg. "Low complexity multiply accumulate unit for weight-sharing convolutional neural networks." IEEE Computer Architecture Letters 16.2 (2017): 132-135. 10.5121/vlsic.2012.3113 153

[2] Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149 (2015).

[3] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[4] Zhang, Chen, et al. "Optimizing fpga-based accelerator design for deep convolutional neural networks." Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays. 2015.

[5] Tang, Song-Nien, and Yu-Shin Han. "A High-Accuracy Hardware-Efficient Multiply–Accumulate (MAC) Unit Based on Dual-Mode Truncation Error Compensation for CNNs." IEEE Access 8 (2020): 214716-214731.

[6] Zhou, Zhi, et al. "Edge intelligence: Paving the last mile of artificial intelligence with edge computing." Proceedings of the IEEE 107.8 (2019): 1738-1762.

[7] Zhou, Zhi, et al. "Edge intelligence: Paving the last mile of artificial intelligence with edge computing." Proceedings of the IEEE 107.8 (2019): 1738-1762.

[8] Du, Li, et al. "A reconfigurable streaming deep convolutional neural network accelerator for Internet of Things." IEEE Transactions on Circuits and Systems I: Regular Papers 65.1 (2017): 198-208.

[9] Medical radiograph compression using neural networks and haar wavelet Muzammil Parvez, M.2016 International Journal of Pharmacy and Technology 8(4), pp. 23499-23507.

[10] Shin, Dongyeob, et al. "Sensitivity-based error resilient techniques with heterogeneous multiply–accumulate unit for voltage scalable deep neural network accelerators." IEEE Journal on Emerging and Selected Topics in Circuits and Systems 9.3 (2019): 520-531.

[11] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energyefficient reconfigurable accelerator for deep convolutional neural networks," IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 127–138, Jan. 2017

[12] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning. nature, 521 (7553), 436-444." Google Scholar Google Scholar Cross Ref Cross Ref (2015).

[13] Sze, Vivienne, et al. "Efficient processing of deep neural networks: A tutorial and survey." Proceedings of the IEEE 105.12 (2017): 2295-2329.

[14] Puri, Raul, et al. "Large scale language modeling: Converging on 40gb of text in four hours." 2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). IEEE, 2018.

[15] Ma, Yufei, et al. "Optimizing the convolution operation to accelerate deep neural networks on FPGA." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 26.7 (2018): 1354-1367.

[16] Bhunia, Swarup, et al. "Hardware Trojan attacks: Threat analysis and countermeasures." Proceedings of the IEEE 102.8 (2014): 1229-1247.

[17] Giacomin, Edouard, et al. "A Multiply-and-Accumulate Array for Machine Learning Applications Based on a 3D Nanofabric Flow." IEEE Transactions on Nanotechnology 20 (2021): 873-882.