Journal of Physics: Conference Series

PAPER • OPEN ACCESS

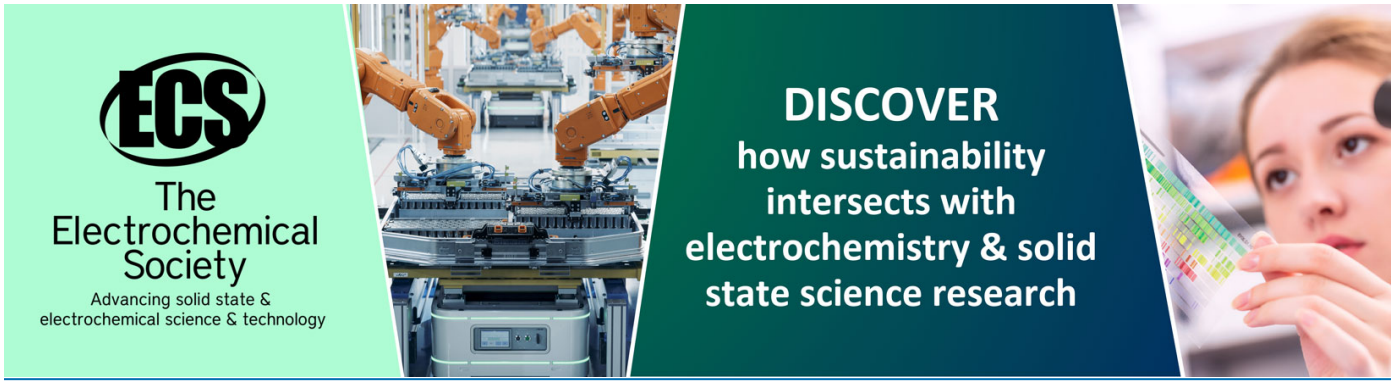
High Level Synchronization and Computations of Feed Forward Cut-Set based Multiply Accumulate Unit

To cite this article: I Srikanth and S Aunmetha 2021 *J. Phys.: Conf. Ser.* **1804** 012201

View the article online for updates and enhancements.

You may also like

- A fast and quantitative measurement of shear wave speed and attenuation using (cross) modal assurance criterion for acoustic radiation force elasticity imaging
Yang Jiao, Chen Yang, Jie Xu et al.
- Multicollision attack on CBC-MAC, EMAC, and XCBC-MAC of AES-128 algorithm
Alfonso Brolin Sihite and Bety Hayat Susanti
- Design and Implementation of High Speed 32-bit MAC Unit
K. Bharghava ram dinesh, R. Vinoth and M.V.R. Kasyap



This content was downloaded from IP address 3.22.181.81 on 07/05/2024 at 19:08

High Level Synchronization and Computations of Feed Forward Cut-Set based Multiply Accumulate Unit

Srikanth I[1], Aunmetha S[2]

[1]Department of Electronics and Communication Engineering,

Methodist College of Engineering and Technology, Hyderabad, Telangana 500001, India
[2]Department of Electronics and Communication Engineering,
KLEF (Deemed to be University), Andhra Pradesh, Guntur District 522 502, India

siri.vlsi@gmail.com

Abstract. In a modern technology-based application, digital signal processing (DSP) is a major priority one, in this gadgets application, the Multiply Accumulate Unit (MAC) will occupy more memory usages, power consumptions and critical path delay. Due to the number of arithmetic operations, this MAC unit will play's a major role in this application product. Thus, the pipelined based architecture will be used to reduce the number of critical paths delay and to improve the performance of MAC architecture. However, the number of flip flops will be increased in the MAC unit, due to number of pipelined architectures. Consequently, it will increase the area and the power consumption. Thus, proposed work of this paper will get a novelty process of feed forward cut-set based MAC architecture with high level synchronization of XOR-MUX full adder with compressor technique. It will reduce the number of logic gates in MAC architecture and hence prove the performance in FPGA Implementation of LUT based area, critical path delay and average power consumption.

Key words— FPGA (Field Programmable Gate Array), LUT (Look-Up Table), HDL (Hardware Description Language), (MFFC) Modified Feed Forward Cutset,(MAC)Multiply Accumulate Unit.

1. INTRODUCTION

In a recent technology, a deep neural network (DNN) will be functional with most of the applications such as automatic speech recognition, medical image analysis, image restoration, military, image recognition, bioinformaticsand toxicology and so on. In this method, a deep and artificial neural network will have an enormous amount of arithmetic operations, such as matrix additions, vector matrix multiplications and so on. In the process of machine learning, the algorithms will accelerator a number of parallel computations, and critical path timing analysis with included and support of multiple accumulate units (MAC). Hence, MAC unit will require a number of logic elements and a number of stages for the multiplication method. For several computation parts, it consists of number of partial product generation and addition method of column and final addition. It will take more stages as well as critical paths, its degradation the performance of MAC unit in DNN network. To minimize this critical path delay and area reduction, various methods have been studied in the research. In this addition method, there will be a number of adders to reduce the critical path delay method such as carry propagation, carry look ahead, and carry select and so on. In the existing method of Wallace tree and DADDA multipliers to achieve fast column addition in number of stages with using carry look ahead (CLA) adder to reduce the critical path delay. Meanwhile, a MAC operation will perform with partial

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

sum operation due to partial product generation, which is generated based on input weight of the multiplication. These, the number of partial productswill increase the number of carry propagation steps, however it will increase the critical path delay [1].

Therefore, the future work of this MAC architecture research will present a pipelined architecture based critical path delay reduction, thus pipelining is a huge popular one to reduced operating clock frequency problems and delay in the architecture, meanwhile this pipelining technique will take more logic gates in VLSI implementation due to number of flip flop will occupied in the architecture and it's also increases the power consumptions of the MAC unit [2]. Therefore, the existing work of this process will introduce feed forward cut set pipelining method as shown example architecture in Fig. 1. It will support to reduced number of logic gates in the architecture after pipelining, thus it will reduce the critical path delay, area and average power consumptions [3].

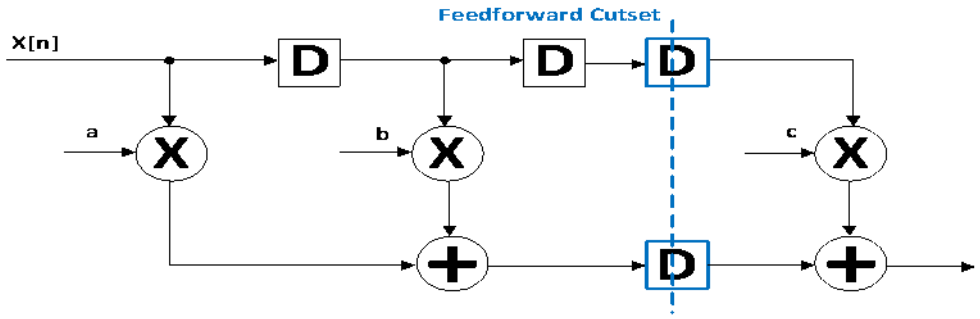


Figure 1: Feed Forward Cut-set based three tap FIR Filter

In this MAC unit architecture, addition is the most important one. After the partial product generation, the additional process will take more effort. In this addition method, there will be different number of operations and different number of architectures such as ripple carry adder, carry look-ahead adder, carry bypass adder, kogge stone adder, and so on. In this different architecture, adders will give the same outputs, however it depends upon the architecture, the logic size will differ. In this case, these all adders will construct using a single conventional full adder; it will take five logic gates for sum and carry generation [4], [5]. Thus, this proposed work will aim to reduce logic gates count in this full adder with using high level synchronization XOR-MUX full adder method. It will have only two logic gates and one multiplexer, it will reduce the logic gates count in all applications with addition supported, and it will reduce the delay of the critical path and power consumption in the architecture. Thus, the main aim of this proposed approach is to minimize the logic gates count and critical path delay in the existing research method of feed forward cut set based on MAC unit [6]. Thus, the proposed approach will introduce a high level synchronization and computation method of feed forward cut set based MAC unit. The MAC architecture will perform many arithmetic operations, mainly this work will focus on the addition process. It will have a number of logic gates, as per the input weight of the multiplication part;these numbers of logic gates will increases.

Therefore, this proposed method will integrate a XOR MUX full adder instead of conventional full adder in the MAC unit with feed forward pipelining method. It will reduce more number of logic gates. Due to this logic gates reduction, logic area, critical path delay and average power consumption will have reduced, and we can prove the better performance in MAC unit architecture [7]. This Feed Forward Cut-Set method will suitable for all type of multiplications such as Vedic Multiplication [20], [21], CSD Multiplier [23]. The majority of this article is set out as follows. Section II will presents level synchronization XOR-MUX Full adder vs conventional full adder, and Section III presents a preliminary

feed forward cut set rule for pipelining architecture, and section IV present proposed FCF & MFCF pipelined architecture, Section V will present FPGA implementation of level synchronization and computation based MAC unit, Section VI will present an evaluation of result and implementation, and finally we concluded this paper in Section VII.

2. Level Synchronization XOR Mux Full Adder vs Conventional Full Adder

In the arithmetic application the adder is the most superior one, it will support many arithmetic operations such as multiplication, address computation, floating point unit, arithmetic logic unit and so on. Thus, the full adder based gate level implementation will have different logic structure, a conventional full adder will have five logic gates for sum and carry generation, in the case a proposed method of XOR gates and MUX based full adder design will occupy only two logic gates and one multiplexer it will take minimum logic area and low power consumptions in arithmetic applications, Fig.2 will shows the circuit diagram of XOR and MUX full adder, and the truth table of XOR MUX full adder design is tabulated in Table.1.In this full adder circuit, it has three inputs structure such as A, B, C in and two output such as SUM and COUT [8].

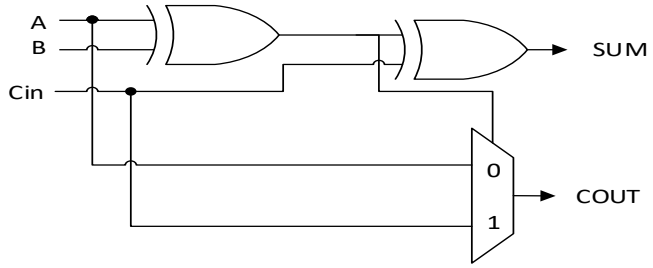


Figure 2 : Circuit diagram of XOR MUX Full Adder

Table 1 : Truth Table for XOR-MUX Full Adder Design

RCA CIN	BEC CIN	A	B	RCA CY	RCA SUM	BEC CY	BEC SUM	CARRY	SUM
0	1	0	0	0	0	0	1	0	0
0	1	0	1	0	1	1	0	0	1
0	1	1	0	0	1	1	0	0	1
0	1	1	1	1	0	1	1	1	0
0	1	0	0	0	0	0	1	0	1
0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0
0	1	1	1	1	0	1	1	1	1

3. Feed forward Cut set rule for pipelining architecture

In the digital signal processing application, a pipelining method was the most important one; it also supports several applications such as image processing and audio processing. The algorithm of this pipelining method will be used to connect a number of flip flops in between two signals; it will decrease the serious path delay and, in that way, increasing the frequency of system clock speed. In the Fig. 1 architecture of the third-order FIR filter will show the examples of this pipelining method [24]. After this pipelining the finite impulse response functioning will be shown in equation (1).

$$y[n] = ax[n] + bx[n-1] + cx[n-2] \quad (1)$$

In this case the pipelining system feed forward cutset eliminates the critical path delay and satisfies the logical equality before and after pipeline. The definition of the feed forward cutset method is as follows. Cutset: A set of the two edges in the architecture at the same time and it will be removed at the same time period. Feedforward-cutset: In the pipelining of two edges, the data will move forward in the same direction. Thus, the Fig. 1 is the valid pipelining architecture of the Feedforward-cutset method, its functioning equality and guaranteed along with the Feedforward-cutset pipelining method.

4. Proposed FCF & MFCF pipelining architecture

The conventional pipelining architecture puts a significant number of flip flops in the FCF architecture, although the crucial route delay is effectively minimized, but the area and power demand will rise due to huge number of flip flops. Thus, the modified Feedforward Cutset (MFFC) would thus substantially eliminate the number of flip flops at traditional pipelined boundary by enhancing the special features of the learning machine algorithm. Fig. 3 displayed the conventional method of 32-bit two stage pipeline accumulatorthat is based on two 16-bit ripple carry adder-based architecture. The input of A[31:0] will give to input buffer, it will stored the data and come apart two part on MSB A[31:16] and LSB [15:0] after that MSB part will give to 16-bit ripple carry adder through inserted flip flop array buffer, and the final output of MSB part will take from the S[31:16] through output buffer [9], [10]. The same operation will be done on LSB portion of A[15:0] but the input directly provided to 16-bit ripple carry adder is configured via the flip flop array and the output buffer of the inserted flip flop array for the time being. At last the whole output of S[31:0] will obtain the entire production.

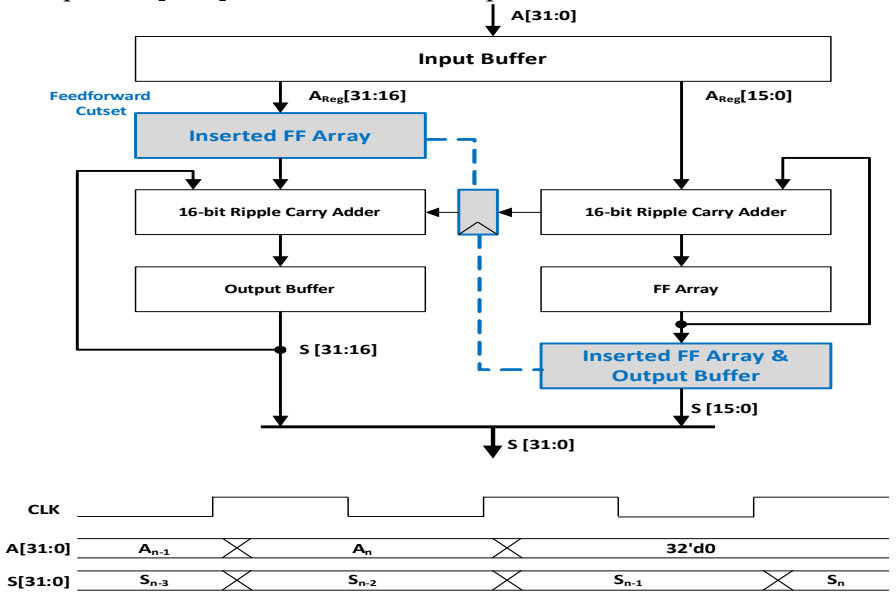


Figure 3 : Conventional Schematic and timing diagram of 32-bit two stage accumulators

The modified FCF-PA based schematic and timing diagram of 32-bit two stage accumulator, will insert one flip flop in between two stage pipelining blocks of 16-bit ripple carry adder. In the traditional PA, as seen in Fig. 3 timing diagram, the appropriate accumulation value of all inputs up to the equivalent clock cycle are generated in every clock cycle, because of the two-state pipeline, and two cycle gap occurs between the input and the subsequent output. In the other hand, only the final consequence of accumulation is true, and the proposed architecture and timing diagram as seen in Fig. 4. Fig. 5 demonstrates examples of how the standard PA and the proposed approach run. The proposed scheme from the lower half of the 32-bit adder [11], [12].

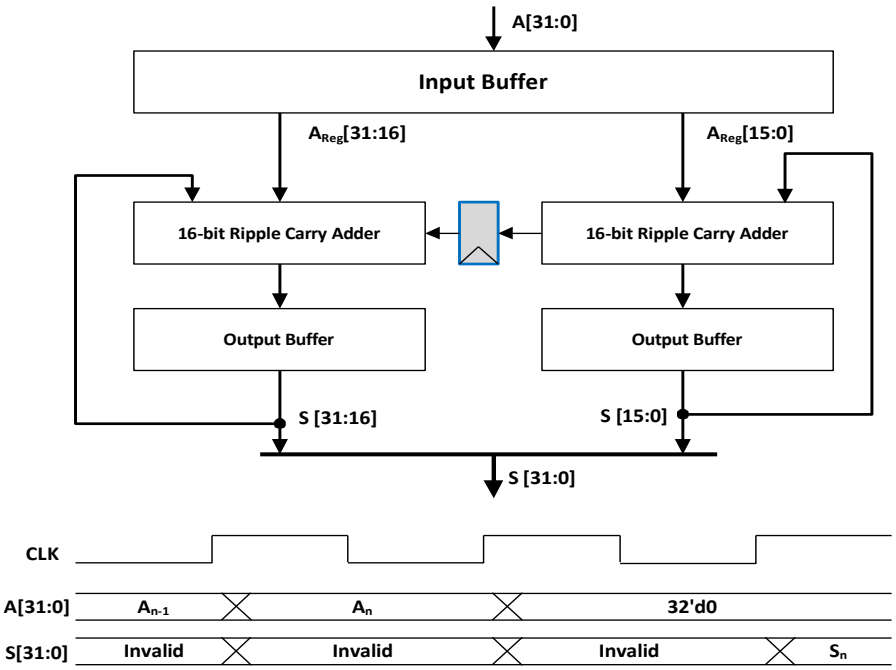


Figure 4 : Modified FCF-PA based Schematic and timing diagram of 32-bit two stage accumulator

The intermediate result is deposited in the proposed accumulator output buffer diverse from the traditional pipelining case outcome. Though, suggested accumulator practices the same output cycle 5. Moreover, the number of cycles from the exclusive input to the final output is same as the two architectures. Mandatory to sum up the features of the suggested FCF pipelining system. The ultimate product weightage is sum of the multiplication between the input function map and the partitions. Meanwhile the CLA was often used to reduce the accumulator's critical path delay [13]. However, the carry theory of projection in the CLA induces a substantial rise in area and power usage.

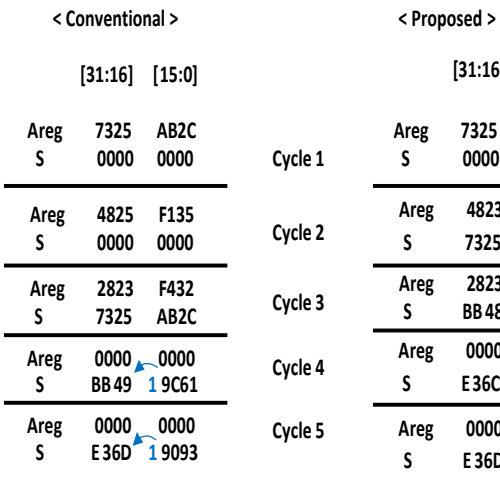


Figure 5 : Examples of two stage 32-bit pipeline accumulation with standard pipelining (left) and FCF-PA suggested (right).

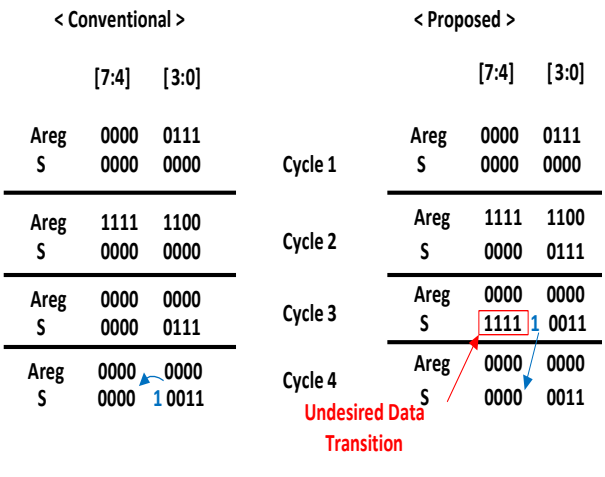


Figure 6 : In the two state 8-bit PA's, an example of the undesired data transfer and input number 2s' complementing 4-bit.

In the traditional pipelining method Fig. 6, in cycle three the accumulation outputs (S) and in cycle two the data stored in the input buffer (AReg) are inserted and stored in the output buffer (S) in Cycle 4. In the FCF-PA, on the other hand Fig. 6, (right), Areg[2], and S[2] are added to generating the carry operation from the cycle 2 [14], [15]. The carry operation was no longer broadcasted into the upper half. In the next step of clock, the carry will be stored in the flip flop. It can be found during the calculation that S[7:4] in cycle 3 shifts to "1111" and in cycle 4 returns to "0000". Even though precision of the accumulation results is not compromised by this undesired data transfer, it decreases the FCF-PA's power performance.

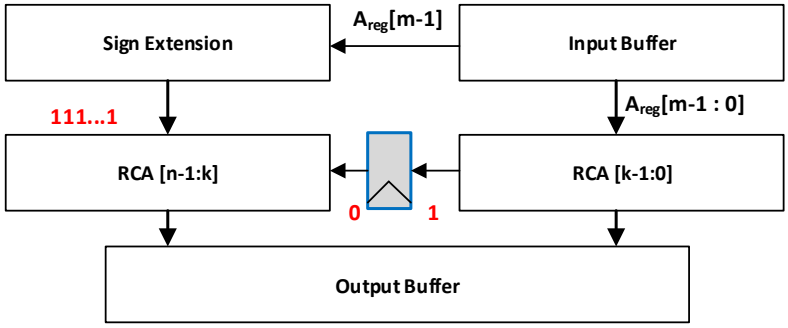
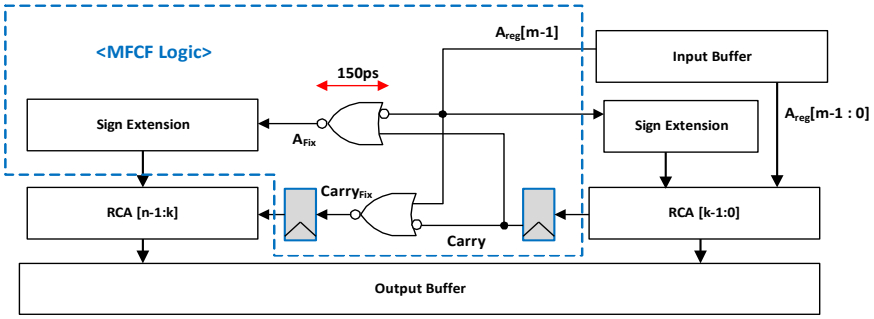


Figure 7 : Existing method of FCF-PA with high power consumptions



A_{reg}[m-1]	Carry	A_{Fix}
0	0	0
0	1	0
1	0	1
1	1	0

A_{reg}[m-1]	Carry	$Carry_{Fix}$
0	0	0
0	1	1
1	0	0
1	1	0

Figure 8 : Modified circuit and improvement of the power efficiency

The efficient variant of the FCF-PA is updated in Fig.8, as the crucial delay is too long if the upper half addition must wait before the result on the lower half condition to happen. Finally, an additional flip flop inserts to avoid the formation of a long critical path. Decreasing the overhead structure, Fig.8 is inserted in one single stage of the pipeline [16]. The remaining stages of the pipeline-PA shown in the Fig.7. The block diagram of MFCF PA is given in the Fig.9. In this good power efficiency shown, even the sign extension bit is reduced, it represents the unwanted data transitions in other RCAs.

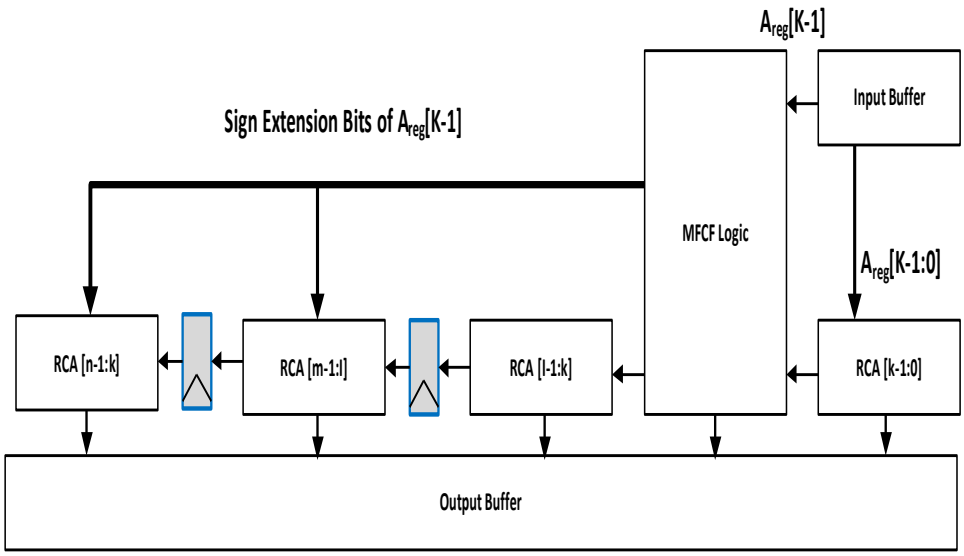


Figure 9 : Block diagram of proposed MFCF-PA

5. FPGA implementation of level synchronization and computation-based MAC unit

In the MAC operations, the column addition using the half adders/full adder in each addition stages for binary numbers is computed. MAC architecture, the special DADDA Multi plierusedin Fig. 10. Many partial products as possible while using the estimation and addition stage of columns in the Wallace tree multiplier. As well as the partial products do not have suitable timing slack to be remove from pipelines, at this case proposed FCF pipelining approach is lower than in the case of the DADDA multiplier [17], [18].

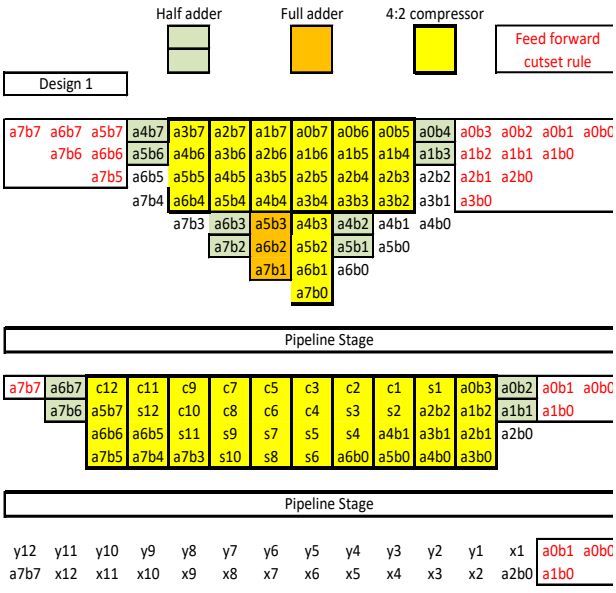


Figure 10 : Proposed FCF-MAC with MFCF-PA using 4:2 compressor method

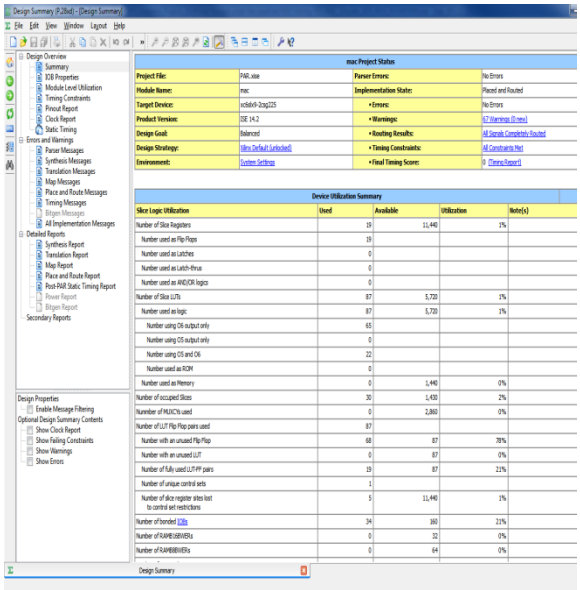


Figure 11 : Synthesis result of FCF-MAC with MFCF-PA

FPGA implementation of this MAC architecture will be carried out by Verilog HDL, the initial method of this implementation to produce several partial products per input weighted multiplication. This proposed work created an 8x8 bit MAC architecture using 64 partial products. After the partial products have been finished, the next step of column addition step 1 will be presented. The performance of column addition stage 1 will be provided to pipeline stage 1, it will be merged with the number of D flip flops, in this case the cutest method of Feed Forward will be implemented to reduce the number of flip flops in this pipeline method. After this stage 1 of pipeline, the sum of the output will be given to stage 2 of column addition, and stage 2 of pipeline. In this column addition process XOR-MUX based full adder will be integrated. Finally, the sums of all these bits given to the proposed MFCF-PA blocks carry addition system. Fig.10 will display the proposed FCF-MAC with MFCF-PA using the 4:2 compressor processes and highlighting the half adder, XOR-MUX full adder, 4:2 compressor and fee forward cutest rule in this number of variable based architecture. Fig.12 (a) will show pipelined based MAC framework architecture with state-of-the-art functionality. Fig. 12 (b) will show the proposed FCF-MAC with MFCF-PA, in this architecture flip flops are removed from the traditional pipelined MAC [19]. Finally, this work is synthesized in Xilinx FPGA and gets the parameter of area results will show Fig. 11, and Delay results will show in Fig. 14. The RTL Schematic of FCF-MAC with MFCF-PA is shown in Fig. 14. Fig.15 will show the power results and Fig.16 will show the simulation result from Modalism Software.

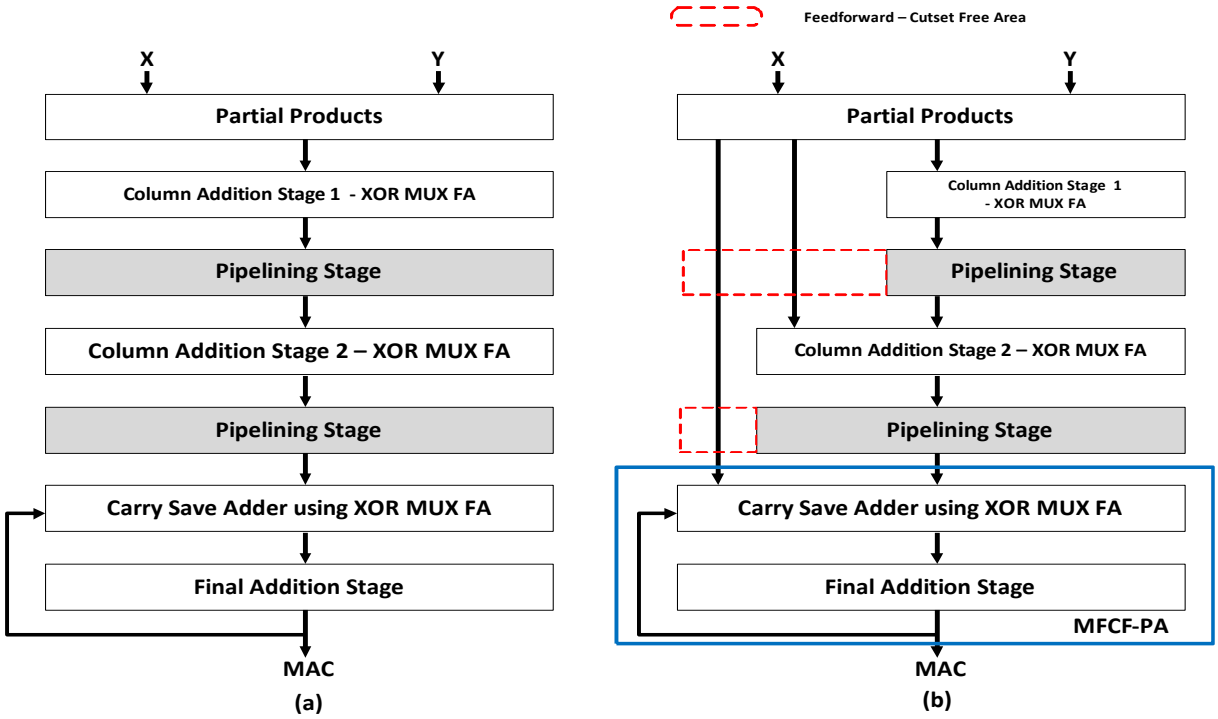


Figure 12 : Flow diagram of Feedforward cutset based MAC architecture (a) Merged MAC with pipelining (b) Proposed FCF-MAC with MFCF-PA

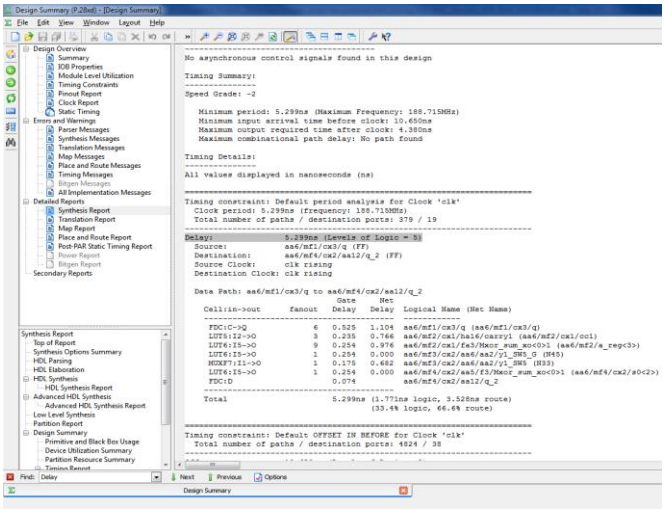


Figure 13: Delay result of FCF-MAC with MFCF-PA

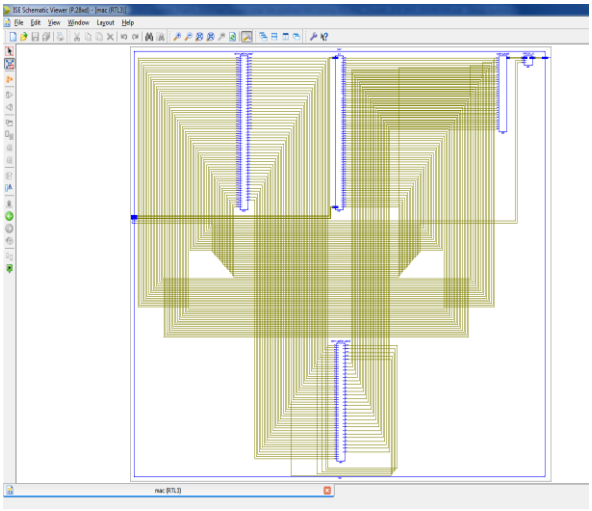


Figure 14: RTL Schematic of FCF-MAC with MFCF-PA

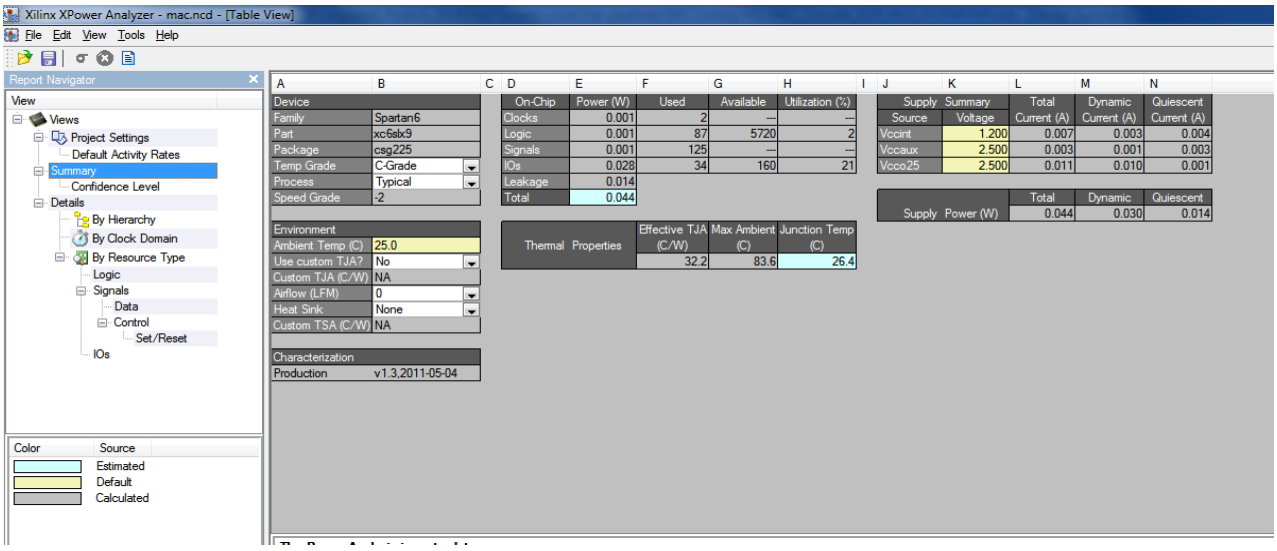


Figure 15: Power result of FCF-MAC with MFCF-PA

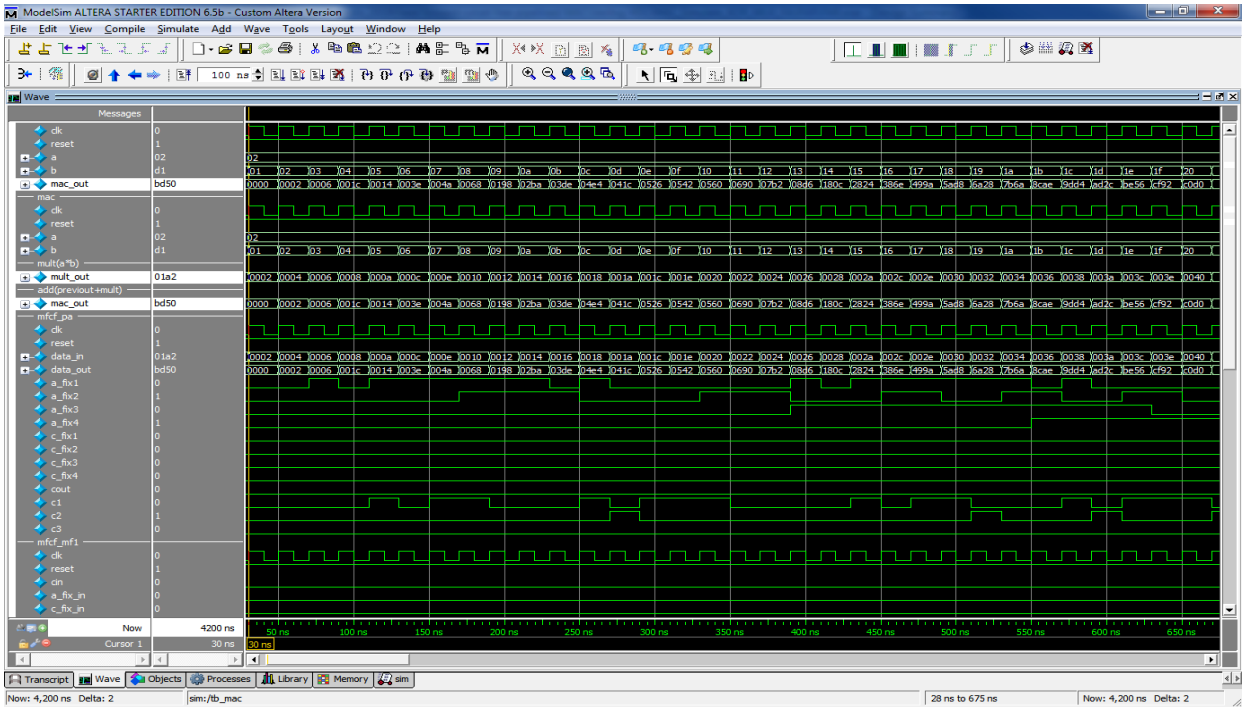


Figure 16: Simulation result of FCF-MAC with MFCF-PA

6. Results and Implementations

In this section, we implement the proposed FCF pipelining process. Only, binary weigh networks are considered for implementation in the accumulator. The 8-bit 2's complement number is used for the input of the accumulator. The design is synthesized with the gate level implementation using Xilinx Design, and different parameter such as Slice register, LUT's, Occupied Slice register, Bonded IOB's, Power and Delay results are obtained. In the Table 2 the comparison of FCF-PA and FCF-PA using XOR-MUX full adder are tabulated, the XOR MUX based FCF-PA will take minimum of 15 LUT's compared to the 105 of existing FCF-PA method. The analysis chart result of FCF-PA and FCF-PA using XOR MUX full adder are shown in Fig. 17. The comparison of proposed MFCF-PA and proposed MFCF-PA with XOR MUX full adder are tabulated in Table.3, and the analysis is shown in Fig. 18. The comparison of MAC using MFCF-PA and MAC using MFCF-PA with XOR MUX full adder are tabulated in Table. 4, in the final MAC comparisons of 8-Bit design, it can be observed the Slice register will take 19LUT's as will take 66 compare to conventional MFCF-PA of 106 LUT's, the proposed method is efficient in Area reduction, the comparison of reduced power and delay of the proposed system is revealed in Fig. 19.

Table 2 : Comparison Table of FCF-PA and FCF-PA using XOR MUX Full Adder

Parameter	FCF-PA	FCF-PA using XOR MUX Full Adder
Slice Register	33	32
LUT's	105	15
Occupied Slice Register	111	5
Bonded IOB's	66	65
Power (W)	0.061	0.014
Delay (ns)	30.275	16.235

Table 3 : Comparison Table of Proposed MFCF-PA and Proposed MFCF-PA with XOR MUX Full Adder

Parameter	MFCF -PA	MFCF - PA with XOR MUX Full Adder
Slice Register	19	19
LUT's	90	87
Occupied Slice Register	34	30
Bonded IOB's	34	34
Power (W)	0.046	0.044
Delay (ns)	5.225	5.299

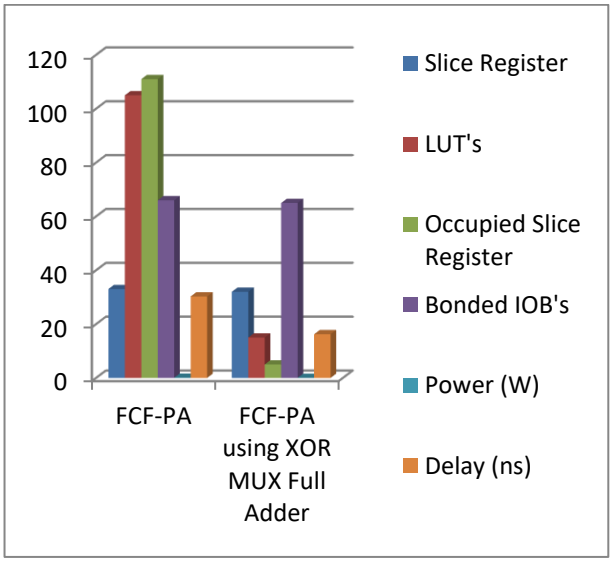


Figure 17: Analysis result of FCF-PA and FCF-PA using XOR-MUX Full Adder

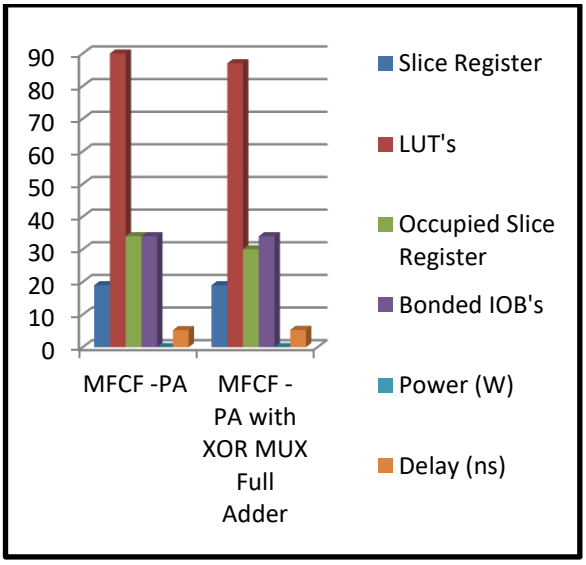


Figure 18: Analysis result of MFCF PA and MFCF PA with XOR MUX Full Adder

Table 4 : Comparison Table of MAC using MFCF-PA and MAC using MFCF-PA with XOR MUX Full Adder

Parameter	MAC using MFCF-PA	MAC using MFCF-PA with XOR MUX Full Adder
Slice Register	19	19
LUT's	106	66
Occupied Slice Register	116	20
Bonded IOB's	34	34
Power (W)	0.047	0.044
Delay (ns)	25.247	24.657

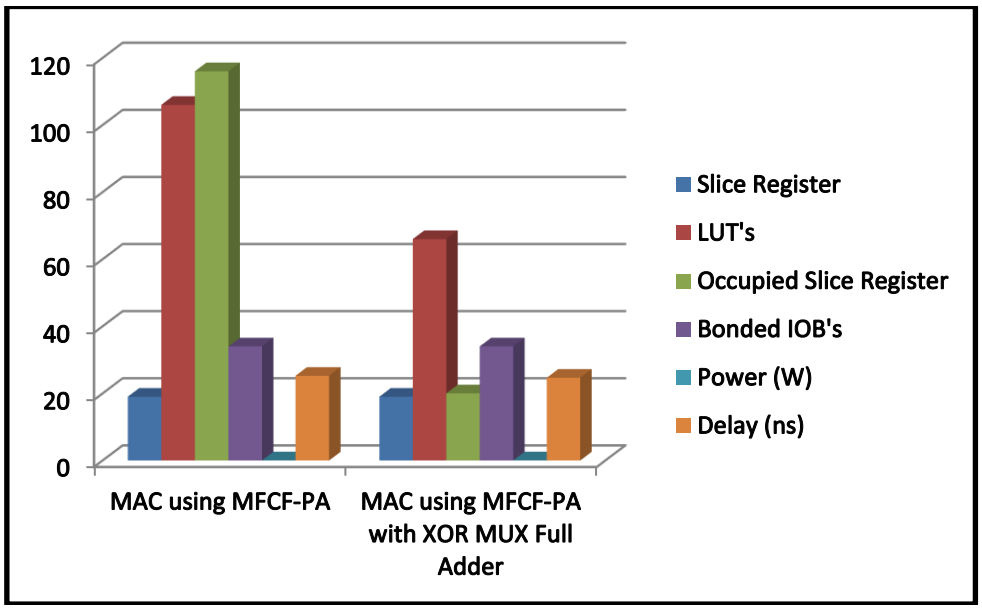


Figure 19 : Analysis result of MAC using MFCF-PA and using MFCF-PA with XOR MUX Full Adder

7. Conclusion

In this proposed article, we presented a novelty-based MAC device with high level synchronization of Feed Forward Cut-Set. The high-level arithmetic synchronization method would reduce the number of logic gates, garbage signal and power consumptions. However, this XOR-MUX maximum adder synchronization would minimize the number of logic gates in the Carry Save Addition in the proposed MFCF-PA block. Due to this decrease, the MAC machine can perform more effectively in optical signal processing related applications. In addition, the proposed work added a 4:2 compressor in column addition of stage 1 and stage 2 architectures to the proposed FCF-MAC with the MFCF-PA block. It has also limited the number of arithmetic operations, such as half adders and full adders. As a result, the proposed MFCF-PA approach would have more field, delay and power output compared to the current FCF-PA based architecture. This Study compared MFCF-PA in two ways: standard full adder based MFCF-PA and XOR-MUX full adder based MFCF-PA. This work was carried out on Verilog HDL and tested with an FPGA device using Xilinx 14.2 at the part number of XC6SLX9-2CSG225, and analyzed the parameter in terms of critical path latency, average power and area.

References

[1] Sungju Ryu, Naebeom Park, and Jae-Joon Kim, Member IEEE, "Feedforward-Cutset-Free Pipelined Multiply Accumulate Unit for the Machine Learning Accelerator", August 2018, IEEE Trans on Very Large Scale Integration.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
[3] K. Simonyan and A. Zisserman. (2014). “Very deep convolutional networks for large-scale image recognition.” [Online]. Available: https://arxiv.org/abs/1409.1556
[4] C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 1–9.
[5] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2013, pp. 6645–6649.

[6] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 577–585.
[7] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 127–138, Jan. 2017.
[8] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.
[9] L. Dadda, "Some schemes for parallel multipliers," Alta Frequenza, vol. 34, no. 5, pp. 349–356, Mar. 1965.
[10] P. F. Stelling and V. G. Oklobdzija, "Implementing multiply-accumulate operation in multiplication time," in Proc. 13th IEEE Symp. Comput. Arithmetic, Jul. 1997, pp. 99–106.
[11] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New Delhi, India: Wiley, 1999.
[12] T. T. Hoang, M. Sjalander, and P. Larsson-Edefors, "A high-speed, energy-efficient two-cycle multiply-accumulate (MAC) architecture and its application to a double-throughput MAC unit," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 12, pp. 3073–3081, Dec. 2010.
[13] W. J. Townsend, E. E. Swartzlander, and J. A. Abraham, "A comparison of Dadda and Wallace multiplier delays," Proc. SPIE, Adv. Signal Process. Algorithms, Archit., Implement. XIII, vol. 5205, pp. 552–560, Dec. 2003, doi: 10.1117/12.507012.
[14] M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 3123–3131.
[15] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in Proc. Eur. Conf. Comput. Vis. Springer, 2016, pp. 525–542.
[16] M. Gao, J. Pu, X. Yang, M. Horowitz, and C. Kozyrakis, "Tetris: Scalable and efficient neural network acceleration with 3d memory," in Proc. 22nd Int. Conf. Archit. Support Program. Lang. Oper. Syst., 2017, pp. 751–764.
[17] A. Parashar et al., "SCNN: An accelerator for compressed-sparse convolutional neural networks," in Proc. 44th Annu. Int. Symp. Comput. Archit., Jun. 2017, pp. 27–40.
[18] Mariya Priyadarshini K., Ernest Ravindran R.S., Atindra Chandra Sekhar M., Sai Kalyan P.J.V., Rahul G. (2019), 'A high-speed precision-controllable approximate 16 bit multiplier', International Journal of Advanced Science and Technology, 28(20), PP.31-41.
[19] Ernest Ravindran R.S., Priyadarshini K.M., Sai A.T., Shiny P., Sabeena S. (2019), 'Design of finite field multiplier for efficient data encryption', International Journal of Advanced Science and Technology, 28(20), PP.42-52.
[20] Venkata Latha G., Syamala Y., Anil Chowdary T., Murali Krishna G. (2019), 'Reversible vedic multiplier', International Journal of Engineering and Advanced Technology, 8(0), PP.852-857.
[21] Sai D.N., Paradhasaradhi D., Ravindran R.S.E. (2019), 'Comparative analysis of efficient hierarchy multiplier using Vedic mathematics', International Journal of Innovative Technology and Exploring Engineering, 8(7), PP.1374-1378.
[22] Priyanka C., Pratap E.R., Bhaskar S., Naresh Y. (2019), 'An optimized area efficient high speed CSD multiplier for image processing applications', International Journal of Innovative Technology and Exploring Engineering, 8(7), PP.1384-1388.
[23] Rooban S., Saifuddin S., Leelamadhuri S., Waajeed S. (2019), 'Design of fir filter using wallace tree multiplier with kogge-stone adder', International Journal of Innovative Technology and Exploring Engineering, 8(6), PP.92-96.
[24] Karunakaran S., Gattim N.K. (2019), 'VLSI implementation of folded FIR filter structures using high speed multipliers', Journal of Engineering and Applied Sciences, 14(4), PP.1070-1077.