International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No.4 (July-2020)

http://dx.doi.org/10.12785/ijcds/090409

Power Efficient Multiply Accumulate Architectures using Modified Parallel Prefix Adders for Low Power Applications

Rakesh S[1,2] and K. S. Vijula Grace[1]

[1] *Department of ECE, Noorul Islam Centre for Higher Education, Thuckalay, Tamil Nadu, India*
[2] *Department of ECE, Mangalam College of Engineering, Ettumanoor, Kerala, India*

Received 29 Jul. 2019, Revised 4 Feb. 2020, Accepted 20 Jun. 2020, Published 1 Jul. 2020

Abstract: This paper presents different power efficient multiply accumulate architectures based on modified parallel prefix adders. A general multiply accumulate unit consists of a multiplier, an adder and an accumulator. The multiplier used in this study is a Vedic multiplier and the parallel prefix adders that are brought into this research include Kogge Stone adder, Brent Kung adder, Han Carlson adder and Hybrid Han Carlson adder. The pre-processing and post-processing stages in those adders which mainly consists of exclusive OR operations are modified using a switch level model of the exclusive OR gate. The corresponding modified adder is also used in the Vedic multiplier for adding the partial products. The performance analysis of the different multiply accumulate models is done in terms of power and figure of merit. The proposed modified multiply accumulate units showed significant improvement in power consumption and figure of merit. The various architectures are designed using verilog hardware description language. The models are simulated and synthesized using Xilinx Vivado Design Suite 2015.2 for Artix-7 field programmable gate array family with xc7a100tcsg324-1 as the target device with -1 speed grade.

Keywords: Brent Kung adder, Han Carlson adder, Hybrid Han Carlson adder, Kogge Stone adder, Multiply accumulate, Parallel prefix adder, Verilog HDL, Xilinx vivado design suite

1. INTRODUCTION

Nowadays portable and handy electronic gadgets are in high demand all over the world. Most of those devices incorporate advanced signal processing capabilities in it. Such devices require signal processors with effective power management system. So VLSI designers are doing extensive research on bringing power efficient portable electronic devices to the external world. While considering to make low power portable electronic devices, precise and prime focus has to be given on designing processors with low power consumption.

The most common unit used in such a digital signal processor is the multiply accumulate (MAC) unit. The unit helps the processor to do multiplication and accumulation process which finds wide range of applications in convolution, filtering, transforms and multiplexing algorithms. The power consumption of the MAC unit determines the power consumption of the processor. The objective of this paper is to design a low power MAC unit by introducing a power efficient parallel prefix adder into the multiply accumulate unit.

A MAC unit discussed in this paper consists of a multiplier, a parallel prefix adder and an accumulator register. The multiplier takes the inputs, produces the partial products and generates the resultant product of the multiplication process. The parallel prefix adder sums up the successive products with the help of the accumulator register. The accumulator register provides the previous result store in it to the parallel prefix adder to carry out the accumulation process. The new resultant is again stored in the register.

To improve the power efficiency of the multiply accumulate unit, the parallel prefix adder which lies in the critical path may be modified so as to reduce the number of components in the adder. This gives rise to the design of a parallel prefix adder with modified pre-processing and post-processing stages with low power consumption. The design results in a slight increase in delay but better figure of merit is maintained.

E-mail: s.rakesh@mangalam.in, vijulasundar@gmail.com

The rest of the paper is organized as follows. Section II presents the review of literatures associated with the research work and section III explains the theory behind a multiply accumulate unit. The modification introduced in the exclusive OR operation is detailed in section IV and section V presents the proposed multiply accumulate architectures using the modified exclusive OR circuit. The simulation results are illustrated in section VI and the conclusion is given in section VII.

2. REVIEW OF LITERATURE

First, Fayez Elguibaly [1] described a multiplication-accumulation merged hardware architecture based on the modified Booth's algorithm. High speed implementation is ensured by employing carry save technique in all the key sections. He focused on four main areas namely the development of a dependence graph for the merged multiply accumulate computation, elimination of delay associated with 2's complement generation in modified Booth algorithm, development of a precise gate delay model and matching the processor word width to the data path width. He also organized the different tasks in the MAC structure as generation of partial products, addition of partial products, final addition and accumulation. Several optimization strategies like generation of carry save 2's complement and breaking the final addition into two stages with first stage having LSB addition and second stage having MSB addition using 2-bit carry look ahead adders, are followed.

Ayman A Fayed et al. [2] explained an architecture for improving the speed of a multiply accumulate unit. The new architecture was based on compressor circuits. He used 4:2 compressor units. The reduction in delay was achieved by feeding the bits of the accumulated value to the unused input lines of the compressor units. This helped to merge the accumulation operation with the multiplication operation thereby saving the need for an additional accumulator. The simulation of the architecture was done in Hspice and found to have improvement in power consumption and speed.

Yuyun Liao et al. [3] proposed a power efficient and high speed multiply accumulate unit. For the efficient handling of media stream, some important features like single instruction multiple data and multiply with implicit accumulate were incorporated in the unit. The features of DSP and multimedia enhancement and double word load allow the efficient handling of media streams. A new fast mixed length encoding scheme is used to achieve high speed and high throughput rate. A combination of complementary pass transistor logic and static complementary metal oxide semiconductor logic is used to achieve low power consumption.

Paolo Zicari et al. [4] introduced an adder accumulator (AAC) architecture for a multiply accumulate unit. The objective was to improve the speed of the unit at the cost of area. The AAC architecture is created by integrating adder to the accumulator register. In this architecture n-bit addition is divided into two (n/2) bit additions. There is a reduction in the delay but the introduction of wait and carry bit registers in the design caused a slight increase in the area. However the results has shown that the modified design provides a valid solution for the problem of carry propagation delay in the implementation of multiply accumulate functions in field programmable gate array.

Tung Thanh Hoang et al. [5] described a two stage pipelined multiply accumulate architecture in which one stage consisted of a circuitry for the generation of partial products and a reduction tree. The second stage solved the sign extension complications. The final carry propagate adder is replaced with a carry save adder along with a new sign extension technique which made the two cycle multiply accumulate architecture faster, energy efficient and area efficient. The introduced architecture gave the multiply accumulate unit the capability to have different operating modes with three modes for multiply accumulate computations and three for multiplication operations. Devika Jaina et al. [6] explained the design of a high speed multiply accumulate unit in which the multiplication is based on vedic mathematics and addition is based on carry save adder. The vedic mathematics is conceptualized from sixteen sutras and the multiplication is done vertical and crosswise manner. The design is coded using VHDL and synthesis is done using Xilinx ISE.

P. Jagadeesh et al. [7] studied the performance of various multiply accumulate unit architectures. It included modified Booth multiplier, Dadda multiplier and Wallace tree multiplier at the multiplier stage and carry look ahead adder, carry select adder and carry save adder at the adder stage. The performance of these models was analyzed in terms of area, delay and power dissipation. The designs were coded using verilog hardware description language and synthesis was done using Cadence RTL compiler. Among all the designs, the model with Wallace multiplier and carry save adder was found to have better performance and the model is used to create higher bit multiply accumulate unit.

Young-Ho Seo et al. [8] proposed a high speed multiply accumulate unit which combined the operation of multiplication and accumulation. The critical path delay is reduced by introducing a hybrid type carry save adder tree thereby improving the output rate. It used radix-2 modified Booth's algorithm for the high speed multiplication. To reduce the number of bits in the final adder carry look ahead addition is incorporated into the carry save adder.

Maroju Saikumar et al. [9] presented different multiply accumulate architecture which uses different multipliers keeping the same carry save adder in the adder stage as in the previous existing system. The various multipliers that are brought to the study are Dadda multiplier, array multiplier, ripple carry array multiplier with row bypass technique, modified radix-2 Booth multiplier and Wallace tree multiplier. The models were designed using verilog hardware description language, simulated and synthesized using Xilinx ISE 13.2 for Virtex – 6 family. The

performance of the models is analyzed in terms of power, speed and area and observed to have optimized performance compared to the existing model.

It is evident from the review of various literatures related to multiply accumulate unit that most of the designs mainly focused on developing high speed and low area architectures. At the same time it is very important to have power efficient designs for a multiply accumulate unit. Such an efficient design of multiply accumulate unit in terms of power consumption will be required for compact and portable devices with processing capabilities.

3. MULTIPLY ACCUMULATE UNIT

The Multiply Accumulate operation is one of the most important computations in various signal processing, filtering, convolution and multimedia application [10-12]. The multiply accumulate unit generally consists of a multiplier, an adder and an accumulator register as shown in Figure 1. The combination of the adder and the accumulator register supports the accumulation process.

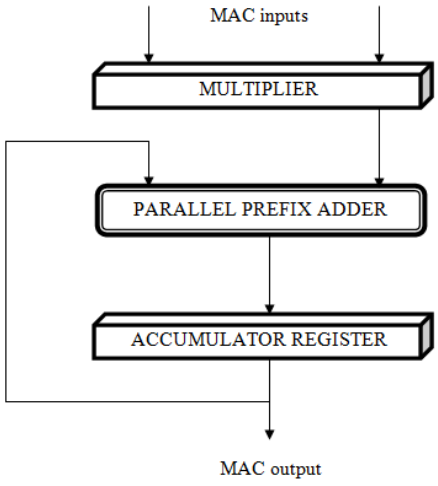

Figure 1. Basic block diagram of a MAC unit using a parallel prefix adder

The multiplier unit takes the N-bit inputs, generates the partial products and adds the partial products to generate the 2N-bit product. The adder block then adds the 2N-bit product with the previous 2N-bit result stored in the accumulator register which can handle 2N+1 bits. This process will lead to accumulation of 2N+1 bits. The accumulator register is normally a parallel in parallel out register which is used to store the result of accumulation. Initially the accumulator register will be cleared and set to zero. The first data that is stored in the accumulator register will be the product of first multiplication process. It is then fed to the adder block to get it added with the result of next multiplication which will lead to the accumulation process and is stored in the register. Thus the multiplication and accumulation process continues based on the number of input signals [13-14]. The multiply accumulate unit tries to realize the expression of the form

$$z[i] = \Sigma\, p[j]\, q[i-j] \quad (1)$$

Here at first the multiplication operation of p and q is performed and without waiting for the availability of next multiplication results, addition is computed in parallel with the multiplication using the multiply accumulate unit. The general expression representing the operation of multiply accumulate unit is

$$Z_i = (P_i \times Q_i) + Z_{i-1} \quad (2)$$

where P_i is the multiplier and Q_i is the multiplicand [15]. For all the addition operations involved in our multiply accumulate units we use modified form of parallel prefix adders for performing addition.

4. PARALLEL PREFIX ADDER (PPA)

A parallel prefix adder (PPA) is a kind of carry look ahead adder which was introduced to reduce the delay occurred in the look ahead technique [16-18]. It is considered to be one of the fastest adders. It has a tree structure and includes three stages of computations in its structure as shown in Fig. 2. The first stage is the pre-processing stage in which the propagate bit and generate bit are calculated. The carry generation is done in the second stage and the third stage, also known as the post-processing stage corresponds to the final sum bit generation [19-21].

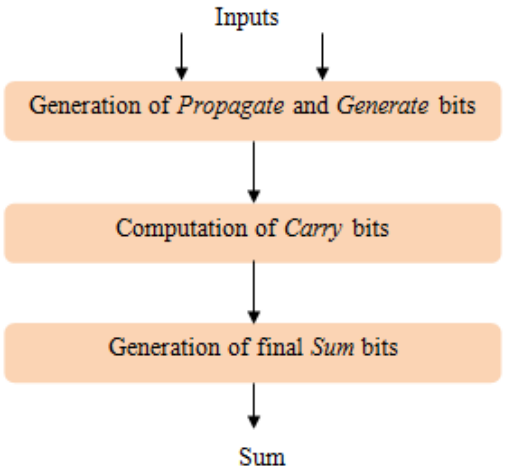

Figure 2. Basic block diagram of a parallel prefix adder

A. *Pre-processing stage*

This stage computes the propagate bit and generate bit as given by the expression (3) and (4)

$$P_i = A_i \text{ xor } B_i \quad (3)$$

$$G_i = A_i \text{ and } B_i \quad (4)$$

where A_i, B_i are the inputs, P_i is the propagate bit and G_i is the generate bit.

B. Carry Computation

The carry calculations are carried out with the help of group generate and propagate (GGP) blocks and group generate (GG) blocks. The GGP block will compute generate bit, G and propagate bit, P as given in the expressions (5) and (6) and its structure is given in Fig. 3.

$$G = G_i + P_i \,.\, G_{previous} \quad (5)$$

$$P = P_i \,.\, P_{previous} \quad (6)$$

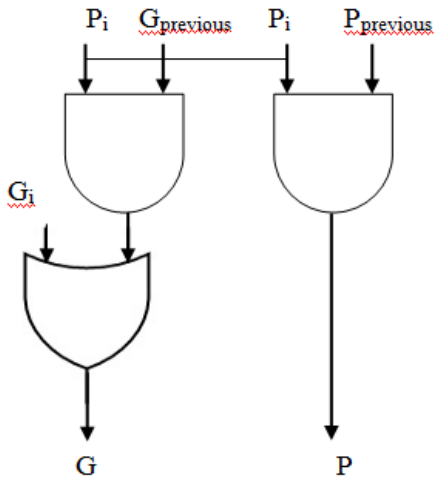

Figure 3. Functions performed by GGP block

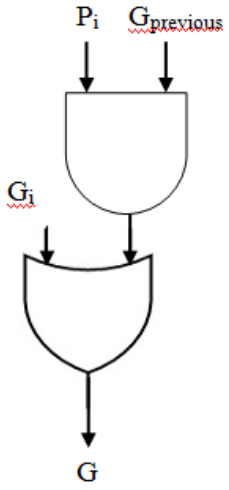

Figure 4. Functions performed by GG block

Fig. 4 shows the function performed by the group generate block. The GG block will compute generate bit, G based on the expression given in (7).

$$G = G_i + P_i \,.\, G_{previous} \quad (7)$$

C. Post Processing

The post-processing stage computes the final sum bit which is given by the expression (8)

$$S_i = P_i \text{ xor } C_{i-1} \quad (8)$$

where S_i is the sum output, P_i is the propagate bit and C_{i-1} is the previous level carry bit.

Some common parallel prefix adders (PPA) that are included in this study are Kogge Stone adder, Brent Kung adder, Han Carlson adder and Hybrid Han Carslon adder [22-26]. A structural comparison of these adders is presented in Table 1.

TABLE I. COMPARISON OF VARIOUS N-BIT PARALLEL PREFIX ADDERS

Type of PPA	Logic Depth	Number of computation nodes
Kogge Stone Adder	$\log_2 N$	$1 + N \log_2 N - N$
Brent Kung Adder	$2 (\log_2 N - 1)$	$2(N - 1) - \log_2 N$
Han Carlson Adder	$\log_2 N + 1$	$(N/2) \log_2 N$
Hybrid Han Carlson Adder	$\log_2 N + 2$	$(N/4) \log_2 N + 0.75N - 1$

5. PROPOSED MODIFICATION IN PPA

It is observed that the main operation handled by the pre-processing stage and the post-processing stage is the exclusive OR operation. The operation is done at the pre-processing stage to compute the propagate bit and at the post-processing stage to compute the final sum bit. In this paper it is proposed to design the exclusive OR operation circuit using a switch level model as given in Fig. 5 which consists of only four transistors instead of conventional CMOS implementation which requires twelve transistors and to use the minimum transistor model in the pre-processing stage and post-processing stage of the parallel prefix adder. The thus modified parallel prefix adder is used in the vedic multiplier for adding the partial products and the same adder is used for facilitating the accumulation process. This will significantly reduce the power consumption of the architecture and improves the power-delay product.

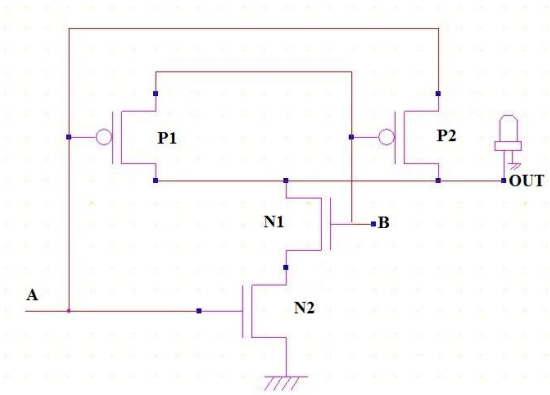

Figure 5. Four transistor model of modified exclusive OR gate

The working of the modified circuit is represented in the Table 2. The status of each pmos and nmos is also presented in the same table.

TABLE II. OPERATION OF SWITCH LEVEL MODEL OF MODIFIED EXCLUSIVE OR

A	B	P1	P2	N1	N2	OUT
0	0	ON	ON	OFF	OFF	$\|V_{tp}\|$
0	1	ON	OFF	ON	OFF	1
1	0	OFF	ON	OFF	ON	1
1	1	OFF	OFF	ON	ON	0

The working of the circuit can be elaborated as follows, When inputs AB = 00, the pmos transistor P1 will be in ON state and P2 will also be in the same state. But the two nmos transistors N1 and N2 will be in OFF state. This is supposed to pull down the output to zero, but due to the poor transmission characteristics of the pmos, the output will be $|V_{tp}|$ [27-30]. When AB = 01, P1 will be ON, P2 will be OFF, N1 will be in ON condition and N2 will be in OFF state. Hence the output will be a high. When AB = 10, P1 will be OFF, P2 will be ON, N1 will be in OFF condition and N2 will be in ON state. So the output will be again a high. So we look at the input and output combinations, it is evident that the circuit performs as an exclusive OR functionality and utilizes merely two pmos transistors and two nmos transistors.

6. PROPOSED MAC ARCHITECTURES

A. MAC with modified Vedic Multiplier and modified Kogge Stone Adder

In this proposed architecture a vedic multiplier performs vertical and crosswise multiplication based on vedic mathematics and generates the partial products. 16 sutras form the basis for this multiplication process. A modified Kogge Stone adder is used in the vedic multiplier for adding the partial products and the same modified adder is used in the critical path for the addition and accumulation process. This modification will result in the reduction of hardware complexity and lead to low power consumption. The proposed architecture is shown in Fig. 6.

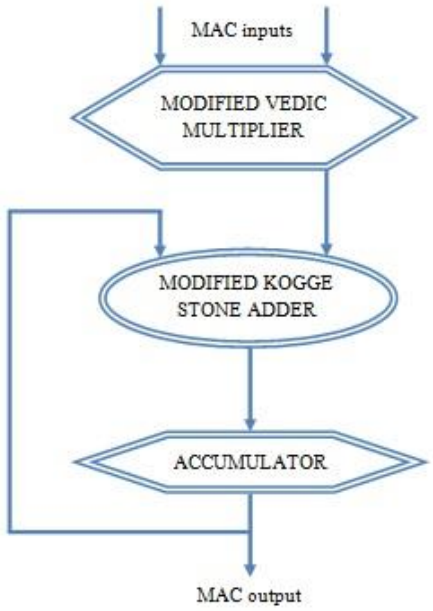

Figure 6. Block diagram of a MAC unit using modified Kogge Stone adder

B. MAC with modified Vedic Multiplier and modified Brent Kung Adder

In this proposed unit a modified Brent Kung adder is used for adding the partial products generated by the vedic multiplier and the same modified Brent Kung adder is used in the second stage which is the adder. This modified Brent Kung adder will contribute to the accumulation process. The result produced is stored in a parallel in parallel out register for providing it for the next accumulation. The proposed structure is presented in Fig. 7.

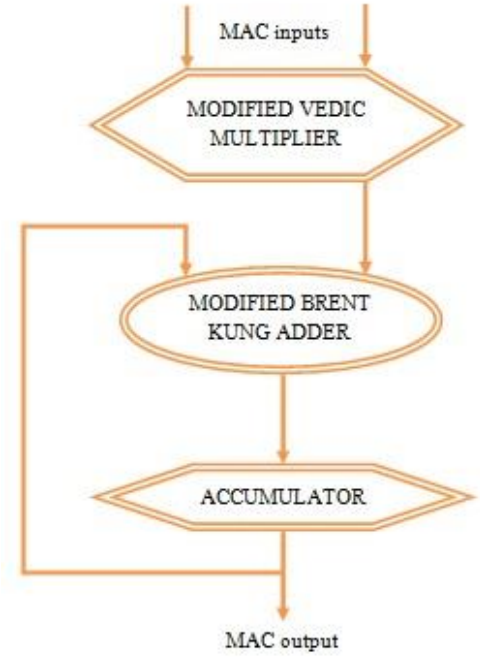

Figure 7. Block diagram of a MAC unit using modified Brent Kung adder

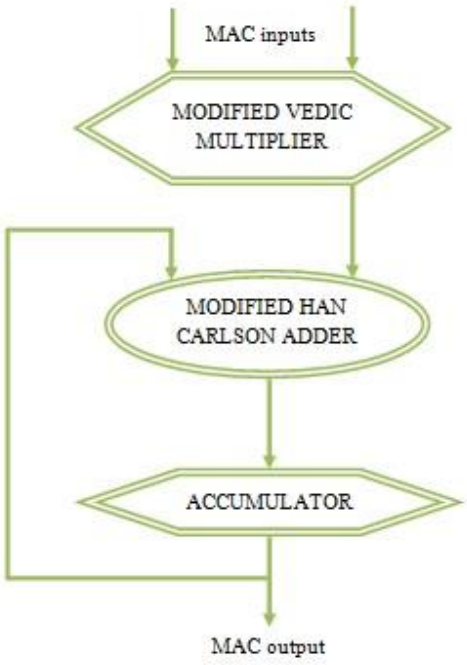

Figure 8. Block diagram of a MAC unit using modified Han Carlson adder

C. MAC with modified Vedic Multiplier and modified Han Carlson Adder

In this proposed structure a modified Han Carlson adder is used in the vedic multiplier and in the adder stage. Han Carlson adder is basically a hybrid parallel prefix adder because it is a combination of Kogge Stone stages and Brent Kung stages. This adder has more number of Kogge Stone stages than the Brent Kung stages. So it is considered to be faster than Brent Kung adder. In Han Carlson adder design the first and last stages are made up of Brent Kung design and all the intermediate stages are utilizing Kogge Stone design. The pre-processing and post-processing stages of the normal Han Carlson adder which mainly involves exclusive OR computation is modified by introducing a four transistor switch level model for performing exclusive OR functionality. This modification results in a modified Han Carlson adder and the same modified adder is used for adding partial products generated as a result of the vedic multiplication process. Also the same modified adder is incorporated at the adder stage of the multiply accumulate unit. The proposed structure is given in Fig. 8.

D. MAC with modified Vedic Multiplier and modified Hybrid Han Carlson Adder

Hybrid Han Carlson adder is also a category of hybrid parallel prefix adder as it contains both Kogge Stone stages and Brent Kung stages. This adder has more number of Brent Kung stages than the Kogge Stone stages. So it has lesser number of computation nodes compared to Kogge Stone adder. In Hybrid Han Carlson adder design the exact middle stages is made up of Kogge Stone design and all the other stages are designed with Brent Kung stages. Modified Hybrid Han Carlson adder is obtained by modifying the pre-processing and post-processing stages of a normal Hybrid Han Carlson adder by introducing a four transistor switch level model for performing exclusive OR functionality in those stages. The modified adder is used for adding partial products in the vedic multiplication process and also the same modified adder is integrated into the adder stage of the multiply accumulate unit. So in this proposed multiply accumulate unit structure a modified Hybrid Han Carlson adder is used both in the vedic multiplier and in the adder stage. The proposed structure is given in Fig. 9.

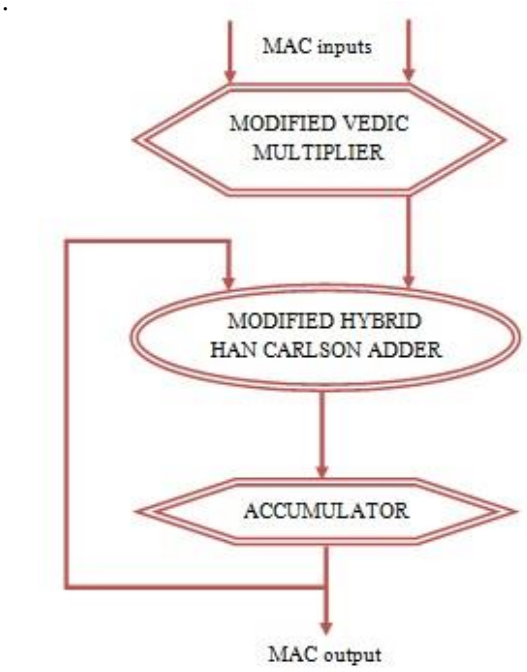

Figure 9. Block diagram of a MAC unit using modified Hybrid Han Carlson adder

The four different proposed multiply accumulate architectures are likely to outperform conventional units in terms of power efficiency. The next section will showcase the performance of the proposed power efficient structures.

7. RESULTS AND DISCUSSION

All the conventional and proposed models are coded using verilog hardware description language. The simulation and synthesis is done using Xilinx Vivado Design Suite 2015.2 for Artix-7 FPGA with xc7a100tcsg324-1 as the target device with a speed grade of -1. The simulation result of the modified exclusive OR is given in the Fig. 10. The result clearly reveals that the modified circuit performs as an exclusive OR gate as it gives a high at the output when the inputs are different and gives out a low when inputs are the same. The simulation waveforms of the four proposed 16-bit MAC units are presented in Fig. 11 to Fig. 14. From these simulation results also it is visible that the MAC unit performs multiplication as well as accumulation process. The inputs are getting multiplied and the result is getting added with the previous result stored in the accumulator register thereby giving out the correct output sequence.

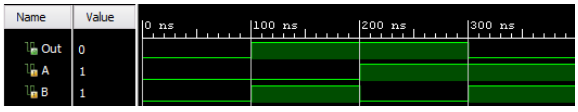

Figure 10. Simulation waveform of modified Exclusive OR

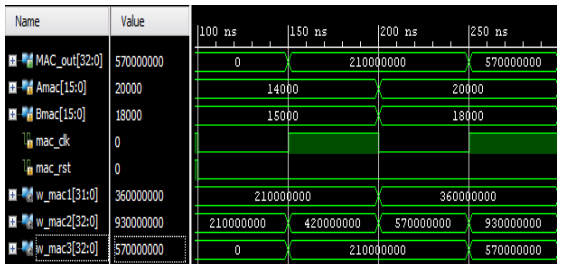

Figure 11. Simulation waveform of MAC unit using modified KSA

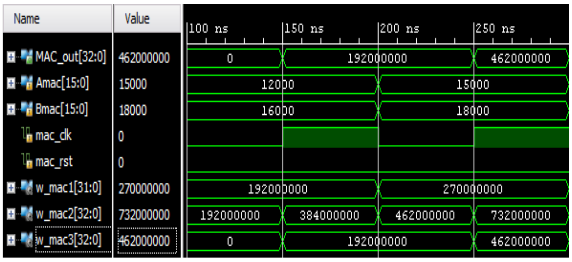

Figure 12. Simulation waveform of MAC unit using modified BKA

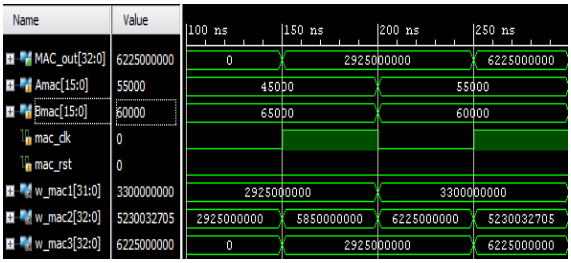

Figure 13. Simulation waveform of MAC unit using modified HCA

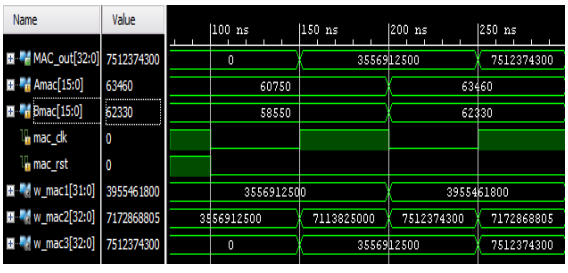

Figure 14. Simulation waveform of MAC unit using modified HHCA

The onchip power is estimated from the power report generated and delay is taken from the timing report. PDP is calculated from the product of power and delay. The estimated values clearly show that there is significant improvement in the performance of the proposed multiply

accumulate units compared to the conventional ones. The same is presented in Table 3.

TABLE III. COMPARISON OF CONVENTIONAL MAC ARCHITECTURES AND PROPOSED MAC ARCHITECTURES

Architecture	Power (mW)	PDP (nJ)	% Saving in Power	% Saving in PDP
MAC with Kogge Stone adder	124	1.473	11.29	6.18
Modified MAC unit with modified KSA	110	1.382		
MAC with Brent Kung adder	120	1.798	10	8.34
Modified MAC unit with modified BKA	108	1.648		
MAC with Han Carlson adder	123	1.430	11.38	6.36
Modified MAC unit with modified HCA	109	1.339		
MAC with Hybrid Han Carlson adder	121	1.770	11.57	9.21
Modified MAC unit with modified HHCA	107	1.607		

The multiply accumulate unit with a combination of modified vedic multiplier and modified Kogge Stone adder showed an improvement of 11.29% in power consumption and 6.18% in the power-delay product. The unit which used modified vedic multiplier and modified Brent Kung adder is observed to have a power saving of 10% and 8.34% improvement in the figure of merit. The performance analysis revealed a 11.38% and 6.36% improvement in power consumption and power-delay product respectively for a multiply accumulate unit with modified vedic multiplier and modified Han Carson adder. It is also observed to achieve a power improvement of 11.57% and figure of merit improvement of 9.21% for the unit with modified vedic multiplier and modified Hybrid Han Carson adder. The comparison of the different MAC architectures in terms of the parameters power and PDP is presented in Fig. 15 and Fig. 16 respectively.

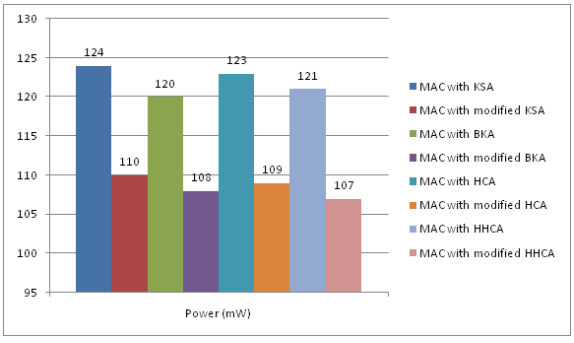

Figure 15. Graphical comparison of power of conventional and proposed MAC architectures

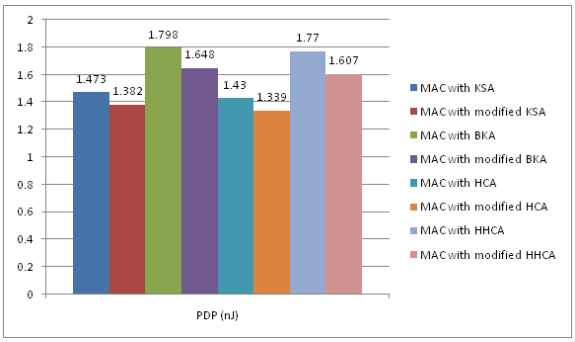

Figure 16. Graphical comparison of PDP of conventional and proposed MAC architectures

8. CONCLUSION

In this paper we proposed four different power efficient modified 16-bit multiply accumulate unit architectures that employed a modified vedic multiplier at the multiplier stage and a modified parallel prefix adder at the adder stage. The modification was done with the objective of reducing the power consumption and improving the PDP which is termed the figure of merit of the device. The parameter estimation clearly revealed that the proposed modification has significant improvement in power consumption as well as the PDP. Implementing our multiply accumulate unit design in a signal processing applications can cause high impact on power efficiency and improved figure of merit for applications that involve intensive multiply accumulate computations. Hence these units find a unique space in low power domain and evolve as a good choice for portable and handy gadgets with signal processing capabilities.

REFERENCES

[1] F. Elguibaly, "A Fast Parallel Multiplier–Accumulator using the Modified Booth Algorithm," IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing., vol. 47, no. 9, pp. 902–908, Sep. 2000.

[2] A. Fayed and M. Bayoumi, "A merged multiplier-accumulator for high speed signal processing applications," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, USA, vol. 3, pp. 3212–3215, May 2002.

[3] Yuyun Liao and David B. Roberts, "A High-Performance and Low-Power 32-bit Multiply-Accumulate Unit with Single-Instruction-Multiple-Data (SIMD) Feature", IEEE Journal of Solid State Circuits, vol.37, no.7, pp.926-931, July 2002.

[4] P. Zicari, S. Perri, P. Corsonello, and G. Cocorullo, "An optimized adder accumulator for high speed MACs," Proc. IEEE International Conference on ASIC, Shanghai, vol. 2, pp. 757–760, Oct. 2005.

[5] Tung Thanh Hoang, Magnus Själander, and Per Larsson-Edefors, "A High-Speed, Energy-Efficient Two-Cycle Multiply-Accumulate (MAC) Architecture and Its Application to a Double-Throughput MAC Unit", IEEE Transactions on Circuits and Systems-I: Regular Papers, vol. 57, no. 12, pp.3073-3081, Dec 2010.

[6] Devika Jaina, Kabiraj Sethi and Rutuparna Panda, "Vedic Mathematics based Multiply Accumulate Unit", Proceedings of IEEE International Conference on Computational Intelligence and Communication Systems, Gwalior, pp.754-757, July 2011.

[7] P.Jagadeesh, S.Ravi, Dr.Kittur Harish Mallikarjun, "Design of High Performance 64-Bit MAC Unit", Proceedings of IEEE International Conference on Circuits, Power and Computing Technologies, Tamilnadu, pp.782-786, 2013.

[8] Young-Ho Seo and Dong-Wook Kim, "A New VLSI Architecture of Parallel Multiplier-Accumulate Based on Radix-2 Modified Booth Alogirithm", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.18, no.2, pp.201- 208, Feb 2010.

[9] Maroju SaiKumar, D. Ashok Kumar and Dr. P. Samundiswary, "Design and Performance Analysis of Multiply-Accumulate (MAC) Unit", Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1084-1089, 2014.

[10] S. Ahish, Y.B.N. Kumar, Dheeraj Sharma and M.H. Vasantha, "Design of High Performance Multiply-Accumulate Computation Unit", Proceedings of IEEE International Advance Computing Conference (IACC), pp. 915-918, 2015.

[11] V. Nithish Kumar, Koteswara Rao Nalluri and G. Lakshminarayanan, "Design of Area and Power Efficient Digital FIR Filter Using Modified MAC Unit", Proceedings of IEEE International Conference on Electronics and Communication Systems, pp. 884-887, 2015.

[12] C.P. Narendra and Dr. K.M. Ravi Kumar, "Low Power MAC Architecture for DSP Applications", Proceedings of IEEE International Conference on Circuits, Communication, Control and Computing (I4C), pp. 404 – 407, 2014.

[13] A. Abdelgawad, "Low Power Multiply Accumulate Unit (MAC) for Future Wireless Sensor Networks", Proceedings of IEEE Sensors Applications Symposium (SAS), pp.129-132, 2013.

[14] Suryasnata Tripathy, L.B. Omprakash, K. Sushanta Mandal and B.S. Patro, "Low Power Multiplier Architectures Using Vedic Mathematics in 45nm Technology for High Speed Computing", Proceedings of IEEE International Conference on Communication, Information & Computing Technology (ICCICT), 2015.

[15] S. Rakesh and K.S. Vijula Grace, "A Survey on the Design and Performance of various MAC Unit Architectures", Proceedings of IEEE International Conference on Circuits and Systems (ICCS), pp. 312 – 315, 2017.

[16] P. Kogge and H. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence relation", IEEE transactions on computers, C-22, pp. 786-793, 1973.

[17] S. Rakesh and K.S. Vijula Grace, "VLSI based Low Power Multiply Accumulate Unit Employing Kogge Stone Adder with Modified Pre-Processing and Post-Processing Stages" in International Journal of Engineering and Advanced Technology, vol. 8, no. 4, pp. 295 – 299, Apr 2019.

[18] R. Brent and H. Kung, "A regulat layout for parallel adders", IEEE Transaction on Computers, Vol. C-31, No. 3, pp 260 – 264, 1982.

[19] S. Rakesh and K.S. Vijula Grace, "Low Power VLSI Design of a modified Brent Kung adder based Multiply Accumulate Unit for Reverb Engines" in International Journal of Recent Technology and Engineering, vol. 7, no. 6, pp. 976 - 980, Mar 2019.

[20] R. Tackdon Han and David A. Carlson, "Fast Area - Efficient VLSI Adders", Proceedings of IEEE 8th Symposium on Computer Arithmetic (ARITH), pp. 49 – 56, 1987.

[21] S. Rakesh and K.S. Vijula Grace, "Modified Han Carlson Adder Based Multiply Accumulate Unit for Low Power Digital Signal Processor" in International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 6, pp. 1144 - 1148, Apr 2019

[22] Sreenivaas Muthyala Sudhakar, Kumar P. Chidambaram, Earl E. Swartzlander Jr., "Hybrid Han-Carlson Adder", Proc. IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), USA, pp 818 – 821, Aug 2012.

[23] S. Rakesh and K.S. Vijula Grace, "Low power multiply accumulate unit based on modified hybrid Han Carlson adder" in International Journal of Engineering and Technology, Vol. 8, No. 1, pp. 1 - 7, 2019

[24] S. Rakesh and K.S. Vijula Grace, "A comprehensive review on the VLSI design performance of different Parallel Prefix Adders" in Elsevier Materials Today: Proceedings, Vol. 11, No. 3, pp. 1001 – 1009, 2019

[25] K.Nehru, A.Shanmugam and S.Vadivel, "Design of 64-Bit Low Power Parallel Prefix VLSI Adder for High Speed Arithmetic Circuits", Proceedings of IEEE International Conference on Computing, Communication and Applications (ICCCA), 2012.

[26] Sudheer Kumar Yezerla and Rajendra Naik, B., "Design and Estimation of delay, power and area for Parallel prefix adders" in IEEE Conference on Recent Advances in Engineering and Computational Sciences (RAECS), 2014.

[27] Binti Mohd Hanib, N., Choong, F., Bin Ibne Reaz, M., Kamal, N. and Badal, T., "Bit Swapping Linear Feedback Shift Register For Low Power Application Using 130nm Complementary Metal Oxide Semiconductor Technology", International Journal of Engineering - Transactions B: Applications, Vol. 30, No. 8, pp. 1126-1133, 2017.

[28] Moallem, P. and Ehsanpour, M., "A Novel Design of Reversible Multiplier Circuit", International Journal of Engineering - Transactions C: Aspects, Vol. 26, No. 6, pp. 577-586, 2013.

[29] Waleed Al-Assadi , Anura P. Jayasumana and Yashwant K. Malaiya, "Pass-transistor logic design", International Journal of Electronics, Vol. 70, No. 4, pp. 739-749, 1991.

[30] A. Askhedkar and G. Agrawal, "Low Power, Low Area Digital Modulators using Gate Diffusion Input Technique". Journal of King Saud University – Engineering Sciences, vol. 31, no. 3, pp. 245-252, July 2019.

Rakesh S received B.Tech degree in Electronics and Communication Engineering in 2008 from Mahatma Gandhi University, Kerala, India. He completed his Masters in Engineering in VLSI Design from Anna University, Chennai, India in 2013. Currently he is pursuing Ph.D in VLSI at Noorul Islam Centre for Higher Education, Thuckalay, India. He has published several research papers in scopus indexed journals. His research work focuses on low power VLSI design and digital VLSI design.

Dr. K. S. Vijula Grace received her B.E degree in Electronics and Communication Engineering from Madurai Kamraj University, Tamil Nadu, India in 1997. She obtained her Masters in Engineering in Power Electronics and Drives from Anna University, Chennai, Tamil Nadu, India in 2005. She received her Ph.D degree under the faculty of Information and Communication Technology from Anna University, Chennai, Tamil Nadu, India in 2015. Her research interests include Embedded systems, VLSI, Communication etc.