FPGA REALIZATION OF LOW POWER MULTI LAYER PERCEPTRON FULL ADDER TO MINIMIZE EDP OF MODULAR MULTIPLIER

Pakkiraiah C and Dr. R V S. Satyanarayana

ABSTRACT

Public key cryptography has been one of the important areas of research for design engineers due to the significance of security in data transmissions. Modular multiplication is part of the most effective cryptographic algorithms. Reduced switching activity factor and cell count are needed in the design of low-power and energy-efficient portable processors for image and digital signal processing, as well as cryptography applications. For public key cryptographic implementations such as digital signature algorithms, RSA algorithms, and ECC algorithms, crypto-primitives are needed to perform modular multiplication on large integers over finite fields. The research for modular multiplier implementations that use less power and have few cells continues, especially for crypto processors, security units, and CPU portable designs. The modular multiplier receives mainly binary input and produces a binary modular product output. This paper introduces a new Multi-Layer Perceptron (MLP) based modular multiplier architecture that uses less dynamic power dissipation. When compared to Proposed Full Adder [1] based Modular multiplier (MM), experimental results shows that the MLP based MM approach can achieve significant reductions in dynamic power dissipation and EDP. In this work, the Xilinx Vivado design suite for the Zynq-7000 family of devices is used to synthesize MLP Full Adder (MLPFA), PFA based MM (PMM) and MLP based MM (MLPMM). Based on implementation results, the MLPMM circuit consumes less dynamic power dissipation than the PFA method. In addition, the implementation results are subjected to a formula-based evaluation in order to determine the design's EDP. When compared to a PFA based modular multiplier, the MLPMM architecture can improve dynamic power dissipation and EDP approximately by 18%.

Index Terms: *Dynamic Power Dissipation, Energy Delay Product, Neural Networks, Cryptography, Full Adder.*

Reference *to this paper should be made as follows: Pakkiraiah C and Dr. R V S. Satyanarayana, (2022), "FPGA realization of low power multi-layer perceptron full adder to minimize EDP of modular multiplier" Int. J. Electronics Engineering and Applications, Vol. 10, No. 2, pp. 01-12.*

Biographical notes:

Pakkiraiah C *received the B.Tech degree from JNTUA, in 2009 and the M.Tech degree from JNTUA in 2013. He is currently a Research Scholar in the Dept. of Electronics and Communication Engineering, SVU College of Engineering, S.V.University, Tirupati. His current research interests include Low Power digital design, Data Security, and VLSI design.*

Dr. R V S. Satyanarayana *is a Professor in the Dept. of ECE, SVUCE. He received Ph.D. degree from S.V.University in 2003. He is currently serving as Principal, SVUCE, S.V.University, and Tirupati. His research interests include Communications and Biomedical Engineering.*

I. INTRODUCTION

In crypto-processors, where the coding and data protection algorithms are combined, the modular multiplier is often employed. The innovative design of full adder is proposed in [1]. As a result, the thorough accomplishment of modular multiplication dispose instructions plays a critical role in the crypto-primitive architecture performance. The innovative discussion of logic minimization methods that objective at decreasing gate count with matching functionality is presented in [2]. A low delay and throughput effective modular multiplier are proposed in [3]. In crypto-processors and information security designs, modular multiplication is a common and widely adopted approach. With the fast development of computational speed and cyber security, the key lengths of crypto processor methods have been significantly increased in order to provide suitable crypto primitive resources. In [4], the novel design of full adder using multilayer perceptron is demonstrated. In [5], the FPGA realization of modular multiplier is developed. Modular multiplication with large digits is the most essential and time-consuming procedure in different asymmetric public-key cryptology. As a result, numerous methods and hardware realization have been proposed to enhance modular multiplication. Configurable based designs are discussed completely modular and are conceived in [6]. The study of digit serial modular multiplication is distinguished in [7]. With the growth of data transfer and high-speed Internet, it's more essential than ever to protect sensitive data. Several symmetric and asymmetric key techniques are utilized to secure private information. In [8], basic logic gates are implemented on FPGA using single layer perceptron. In [9], high performance hardware realization of a modular multiplier utilizing a RN is presented. Cryptography protects information transmission over an entrusted network even while preserving the veracity and privacy of the transmitted information. Various crypto-primitive techniques are utilized to encrypt and decrypt information at the origin and destination, providing data privacy. A well-organized realization of modular multiplication is reported in [10]. To minimize the critical path delay, truncated modular multiplication is proposed in [11]. The Residue Number System (RNS) is a unique number categorization that has recently been studied in order to achieve high-speed, low-power dissipated devices. Large size numbers are composed of small remainders in RNS, which are achieved by the use of forward controllers. In [12], the detail explanation of neural network configurations is reported. In [13] switching activity concept is introduced to minimize dynamic power dissipation. Asymmetric public-key cryptographic algorithms, such as DSA and RSA, depend on modular multiplication. Modular multiplication in DNS has gained popularity in recent years due to its rapid arithmetic methods on big operands. Adders are categorized based on Boolean logic expression realization under various power supply voltages and load values in [14]. To minimize switching activity factor, k-map approach is addressed in [15]. Electronics advancement in recent decades have made modern computers more accessible to the general public. Crypto-primitives must be established, however, to guarantee the secure transmission of confidential documents over a public network. Hardware implementations especially on digital circuits allowing for more efficient computations with less power dissipation are required. Two different carry save addition algorithms are reported to reformulate modular multiplication in [16]. The scalable module is discussed for computing modular multiplication in [17]. In current engineering, such as gas and oil, and fragile organizations, such as private transportation and power electronic systems, secure communication is a significant issue. Confidentiality is the most basic piece of equipment used to authentication and encryption in these systems. In [18], an extension of scalable architecture is presented to compute modular multiplication. In [19], the minimization of switching activity factor in combinational circuits is reported. Cryptography primitives have been structured in unique and extensive ways in

recent years to construct creative and controlling digital security mechanism, such as distinctiveness cryptographic. This situation has prompted the development of traditional hardware cryptosystems to perform these modular processes more quickly and improve the overall performance of high-level data security solutions. The modular multiplication operation is extended to compute division and inverse operation for power of binary modulus is proposed in [20]. The problem of module size is overcome using special crypto primitives are introduced in [21]. The following has formed the basis of this research, which is a significant extension of [1]. Section II provides sufficient prerequisites for recent research initiatives. The design of full adder logic circuits using PFA and Multi-layer perceptron is explored in Section III. Section IV discusses the development of modular multiplier using PFA and multi-layer perceptron. The simulation and implementation results of designs are shown in Section V, followed by the conclusion.

2. PRELIMANARIES

A. Power dissipation

The supply voltage (Vdd) and leakage currents in a Metal Oxide Semiconductor (MOS) are proportional to the static power dissipation.

$$P_{static} = I_{leakage} \times V_{dd} \tag{1}$$

The switching activity, capacitive loads, operating frequency, and supply voltage all contribute to the dynamic power dissipation of MOS devices (Vdd).

$$P_{dynamic} = \alpha \times C_L \times f \times V_{dd}^2 \tag{2}$$

Switching activity factor accounts for 80% of power dissipation in today's technologies. It is vital to reduce the SA value of digital Very Large-Scale Integration (VLSI) circuits in order to minimize power loss in MOS circuits.

B. Figure of Merit

For digital circuits, the product of the input to output propagation delay and the average power consumption has been employed as the FOM. FOM is a parameter for evaluating and comparing devices used in linear circuits. The FOM method is used to estimate the quality of a digital logic gate.

$$FOM = P_{avg} \times t_{pd} \tag{3}$$

C. Power Delay Product

The purpose of a digital circuit designer is to reduce PDP as much as possible in order to achieve low power in frequency- operated digital circuits. The PDP is a measurement of the gate's energy usage. The PDP stands for per-switching-event dynamic power consumption. The quantity of energy required to do the computation is measured by the power delay product. A logic circuit with a minimal PDP, on the other hand, may perform computations exceedingly slowly.

$$PDP = P_{dynamic} \times t_{pd} \tag{4}$$

D. Energy Delay Product

As a result, the energy delay product is estimated to comment on the digital logic circuit's good performance. The product of the power delay product and the input to output propagation delay is the energy delay product of any digital circuit.

$$EDP = PDP \times t_{pd} \quad (5)$$

3. ADDER DESIGNS

A. Proposed Full Adder (PFA)

The Proposed Full Adder (PFA) [1] is made up of three modules. XOR logic is used to represent module 1 and module 2. The Carry Generation Network (CGN) module 3 is utilized to create the output carry bit. The proposed full adder design is built on intermediate results (P), which is defined as the signal P between A and B inputs. The sum and carryout expressions can be obtained using the intermediate result, and they are represented as

$$P = \overline{AB}\,(A + B) \quad (6)$$

$$S = \overline{PC}\,(P + C) \quad (7)$$

$$C_0 = PC_i + AB \quad (8)$$

The logic diagram of proposed full adder is shown in Figure.1

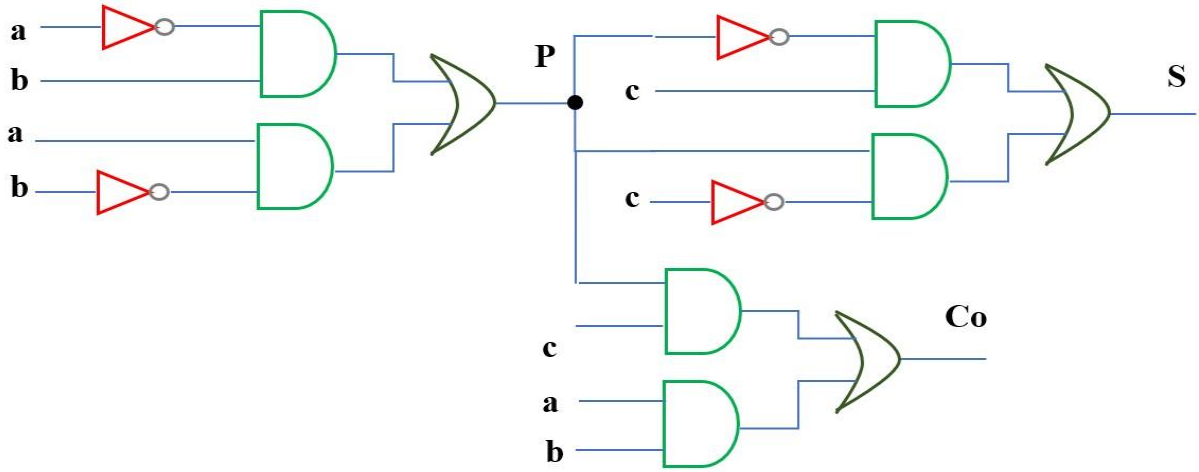

Fig.1. Logic diagram of PFA

B. Full Adder Design using Multi-Layer Perceptron (MLPFA)

The XOR gate is also designed using Multi-Layer Perceptron (MLP) neural network method then it is named as MLP XOR. The Figure.2 shows the representation of MLP XOR gate. The basic mathematical equations of MLP XOR gate is given below

$$H1 = A \times w1 + B \times w3 \quad (9)$$

$$H2 = A \times w2 + B \times w4 \quad (10)$$

$$y1 = \begin{cases} 1, & H1 > -3 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

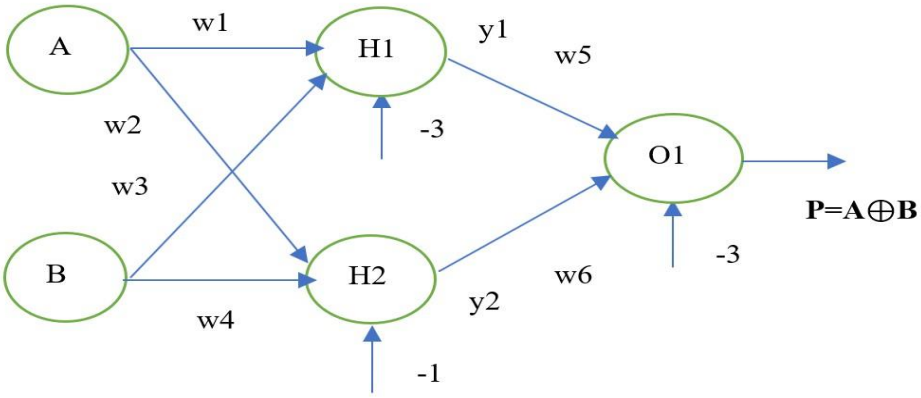

Fig.2. XOR gate using MLP method

$$y2 = \begin{cases} 1, & H2 > 1 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$$O1 = y1 \times w5 + y2 \times w6 \tag{13}$$

$$P = \begin{cases} 1, & O1 > 3 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

The functional table of two input XOR gate using multi-layer perception logic is shown in Table I.

Table I. Functional table of XOR gate using MLP

a	b	w1	w2	w3	w4	H1	H2	y1	y2	w5	w6	O1	P
0	0	-2	2	-2	2	0	0	1	0	2	2	2	0
0	1	-2	2	-2	2	-2	2	1	1	2	2	4	1
1	0	-2	2	-2	2	-2	2	1	1	2	2	4	1
1	1	-2	2	-2	2	-4	4	0	1	2	2	2	0

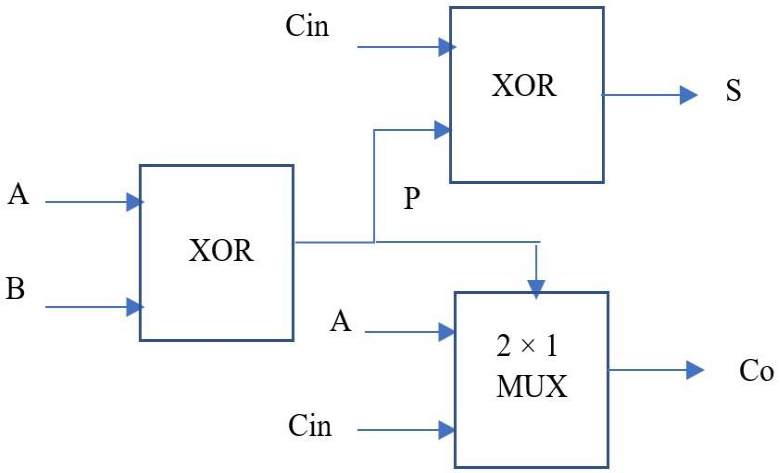

Fig.3. Block diagram of PFA

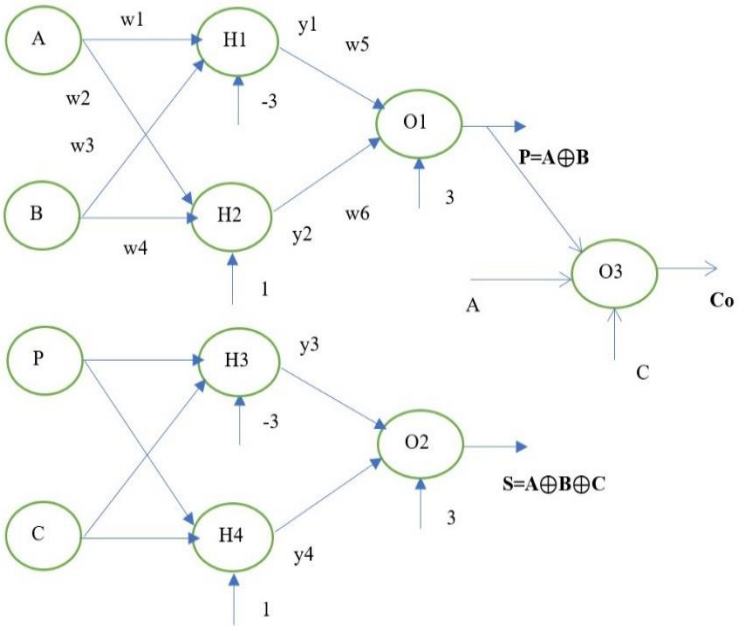

Fig.4. ANN model diagram of PFA

The block diagram of proposed full adder is shown in Figure.3. The XOR gates used in the design of proposed full adder are modeled using multi-layer perceptron method. The ANN model of proposed full adder is shown in Figure.4.

4. DESIGN OF MODULAR MULTIPLIER

A. Design of MM using PFA (PMM)

Arithmetic logical operations over the finite field Zp = [0, 1, 2 ,. . .. . . |p-1|] are used as ciphering primitives for executing various cryptographic algorithms. The basic operation used in cryptographic algorithms is modular multiplication. Given two integers X and Y, modular multiplication computes Z=(X×Y) mod m. The architecture for computing Z value is shown in Figure.5. The PFA based MM unit consists of parallel register, two m-bit PFAs, multiplexer (M), Shift register, counter and control unit.

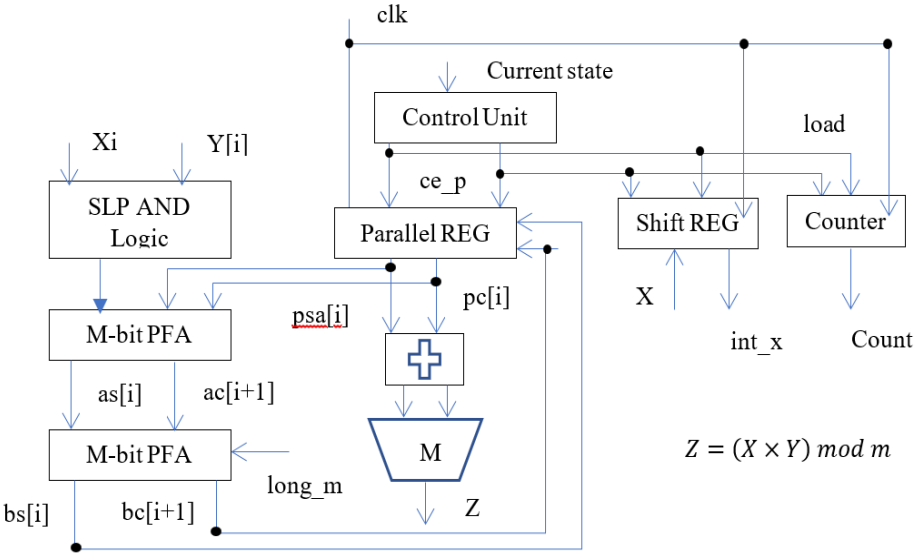

Fig.5. Block diagram of MM using PFA.

The control unit is used to find the load and ce_p value based on the clock and next state values, which depends upon start, equal_zero, reset and clock signals. Based on the current state value, the load, update and done logic values are generated using control unit2. From the load, cep and clock signals counter determine the count of clock pulses to complete the Z=(X×Y) mod m. The parallel register has a bs[i], bc[i+1], load, ce_p and clock signals as inputs and redirects to two outputs (psa[i] and pc[i]). The SLP based m-bit adder performs the addition operation between XiY[i], psa[i] and pc[i]. The SLP modular multiplier is realized by using AND logic units and m-bit PFAs without changing its functionality. The block diagram of PFA based modular multiplier is shown in Figure.5. The RTL diagram of modular multiplier using PFA is shown in Figure.6.

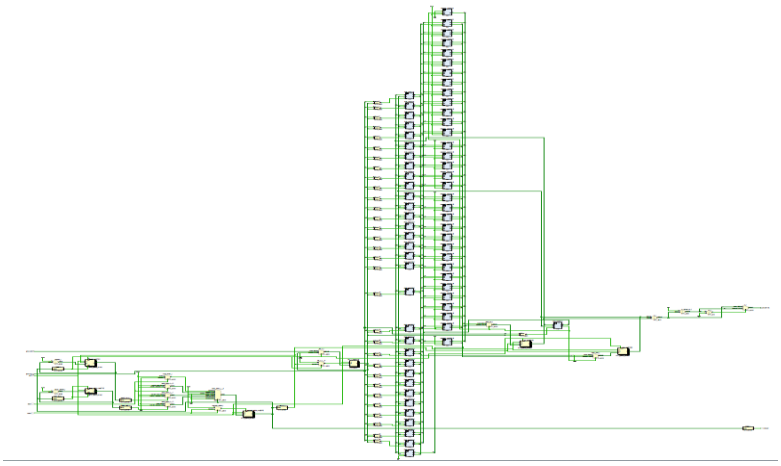

Fig.6. RTL diagram of MM

B. Design of MM using MLPFA (MLPMM)

Without modifying its functionality, the MLP modular multiplier is realized using AND logic units and proposed m-bit MLP full adders. The block diagram of PFA based modular multiplier is shown in Figure.7. The state machine of modular multiplier is shown in Figure.8.The modular multiplier was originally set to the standard state, with the raise start signal followed by the load signal, which loaded the x value into the shift register and cleared the parallel registers.

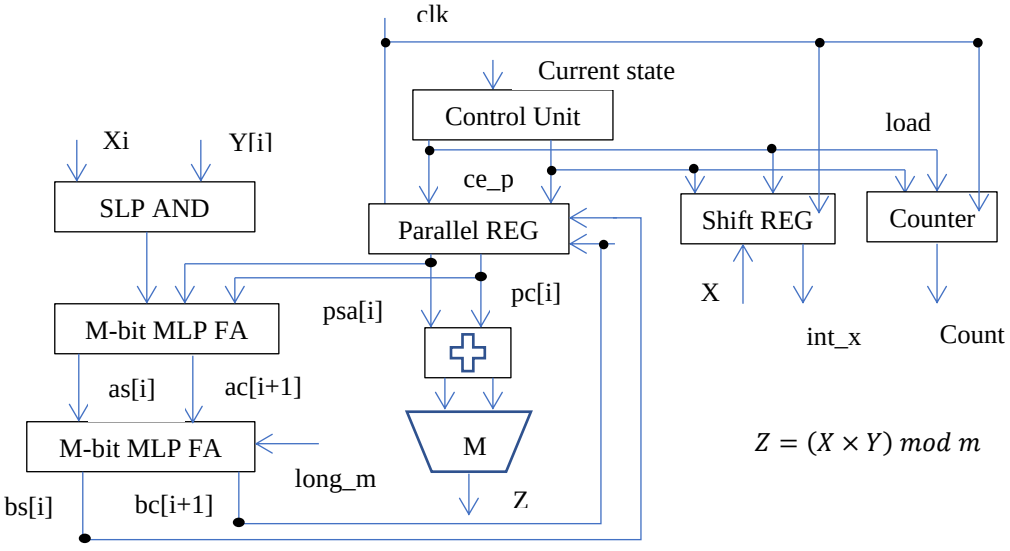

Fig.7. Block diagram of MM using MLPFA

A counter counts up to the length of x's bit length (192-bit). The shift register is shifted every clock cycle, and the second and third lines of the aforementioned algorithm are executed. When the counter hits 192, a timer is used to reset the counter to zero, allowing the last line of pseudo code to be executed. Transmit the done signal when the timer reaches zero.

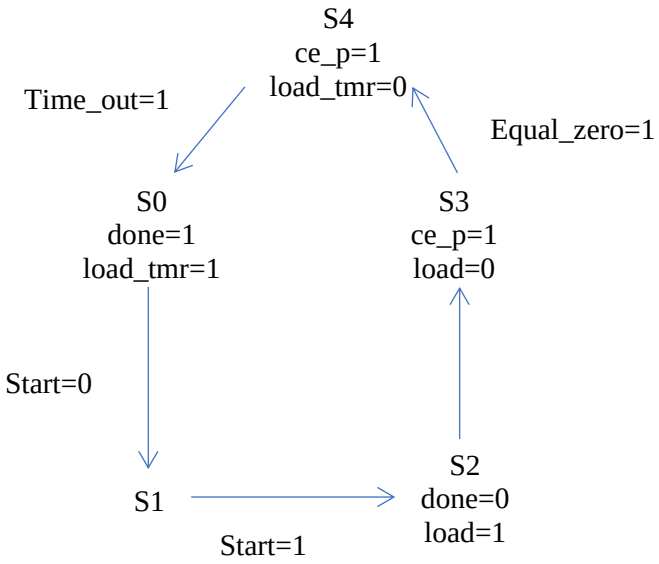

Fig.8. State machine of Modular Multiplier

5. SIMULATION AND IMPLEMENTATION RESULTS

Xilinx Vivado behavioral simulation is used to verify the functionality of the full adder and modular multiplier circuits. The simulation results for two test vectors, 101 and (X=F7 and Y=A), are shown below. The output waveforms for the two input combinations are displayed in the Figure.9. The outputs of the simulation tool are physically tested on the FPGA ZYBOZ7. The Verilog code is synthesized and implemented using Xilinx Vivado Zynq 7000 target device configurations for this process. After the implementation process, the net list is established. The implementation results of MLPFA, PMM, and MLPMM are shown in Figure.10. From Figure.10 we observe that, the proposed MLP Full adder circuit consumed a total power of 155mW with a propagation delay of 6.567nsec, the MM using PFA circuit consumed a total power of 1.712W with a propagation delay of 16.840nsec, and the MLP based MM architecture consumed a total power of 1.418W with a propagation delay of 17. 11nsec.The Table II shows the overall comparison of PFA, MLPFA, PMM, and MLPMM in terms of static power, dynamic power, and delay.

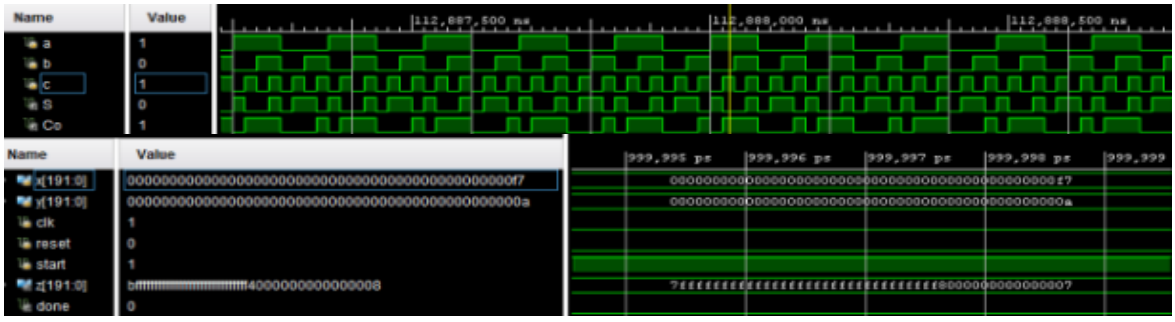

Fig.9. Simulation results of full adder and modular multiplier.

It becomes performance measurements of the proposed circuits after applying the formula-based evaluation on implementation results. We concentrated on the energy delay product of all designs in this research. The proposed method outperforms the current method in terms of the energy delay product of PFA, MLPFA, PMM, and MLPMM. All of the designs' performance metrics are presented in a Table III. According to the Table III, the MLP method outperformed the conventional method in terms of Figure of Merit, power delay product, and energy delay product. The performance metrics of PFA, MLPFA, PMM, and MLPMM in terms of percentage is estimated between PMM and MLPMM based on dynamic power dissipation and EDP. From this estimation, the dynamic power dissipation and EDP of MLPMM is improved by approximately 18% over PMM design.

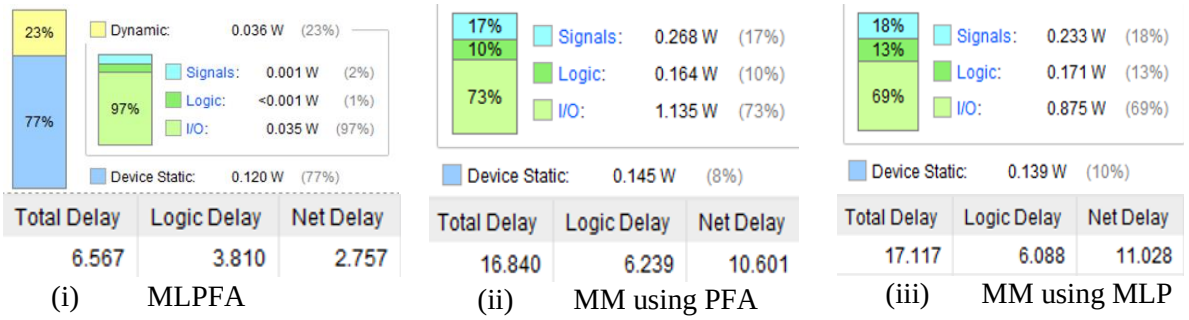

Fig.10. Synthesis results of (i) MLPFA (ii) PMM (iii) MLPMM

Table II. Synthesis results of MM designs

Name of the Design	Ps (W)	Pl (W)	Pio (W)	Pstatic (W)	Delay (nsec)
PFA	0.001	<0.001	0.089	0.121	7.016
MLPFA	**0.001**	**<0.001**	**0.035**	**0.120**	**6.567**
MM using PFA	0.268	0.164	1.135	0.145	16.840
MM using MLPFA	**0.233**	**0.171**	**0.875**	**0.139**	**17.117**

Table III. Performance metrics of MM designs

Name of the Design	Pavg (W)	Pdynamic (W)	FOM (nJ)	PDP (nJ)	EDP ($\times 10^{-18}$ J)
PFA[1]	0.105	0.090	0.749	0.639	4.54
MLPFA	**0.078**	**0.036**	**0.512**	**0.236**	**1.54**
MM using PFA	1.567	0.164	14	26	437
MM using MLPFA	**1.279**	**0.171**	**12**	**21**	**359**

6. CONCLUSION

In this work, we offer a modular multiplier with lower cell count and EDP than existing designs while having lower dynamic power dissipation. When it comes to implementing crypto basic algorithms on a hardware platform, modular multiplication is crucial. The logic decomposition technique was intended to lower the modular multiplier's power dissipation even more. The proposed modular multiplier can also be utilized to implement DSA with fast arithmetic. This method minimizes the number of dynamic logic gates used in data calculation and guarantees a reduction in arithmetic architectural design. When compared to a PFA based modular multiplier, the MLPMM architecture improved dynamic power dissipation and EDP approximately by 18%. We just design the MLP based modular multipliers in this study, leaving physical realization and simulation to be

discussed later. The proposed modular multiplier is also quite adaptable and can be utilized well in a security processor. As a result, MLP based modular multiplier design can be applied to a wide range of real-time applications, including CPUs, crypto processors, and security systems. The MLP based low-power modular multiplier can be used in embedded devices with limited power resources to create low-power crypto processors.

REFERENCES

[1] C. Pakkiraiah and D. R. Satyanarayana, "An innovative design of low power binary adder based on switching activity," International Journal of Computing and Digital Systems, vol. 11, no. 1, pp. 861–871, 2022.

[2] S. Raghuraman and L. Nazhandali, "Does gate count matter? hardware efficiency of logic-minimization techniques for cryptographic primitives".

[3] A. Parihar and S. Nakhate, "Low latency high throughput Montgomery modular multiplier for RSA cryptosystem," Engineering Science and Technology, an International Journal, 2021.

[4] R. Ganesh and D. B. Prakash, "FPGA realization of logic gates using neural networks", CVR Journal of Science and Technology, vol. 20, no. 1, pp. 61–66, 2021.

[5] A. A. Abd-Elkader, M. Rashdan, E.-S. A. Hasaneen, and H. F. Hamed, "Advanced implementation of Montgomery modular multiplier," Microelectronics Journal, vol. 106, p. 104927, 2020.

[6] V. Bianchi and I. De Munari, "A modular Vedic multiplier architecture for model-based design and deployment on FPGA platforms", Microprocessors and Microsystems, vol. 76, p. 103106, 2020.

[7] S. Fatemi, M. Zare, A. F. Khavari, and M. Maymandi-Nejad, "Efficient implementation of digit-serial Montgomery modular multiplier architecture," IET Circuits, Devices & Systems, vol. 13, no. 7, pp. 942–949, 2019.

[8] R. Sabbaghi, L. Dehbozorgi, and R. Akbari-Hasanjani, "New full adders using multi-layer perceptron network," International Journal of Smart Electrical Engineering, vol. 8, no. 03, pp. 117–122, 2019.

[9] Y. Kong and M. S. Hossain, "FPGA implementation of modular multiplier in residue number system," in 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS). IEEE, 2018, pp. 137–140.

[10] S. Khan, K. Javeed, and Y. A. Shah, "High-speed FPGA implementation of full-word Montgomery multiplier for ECC applications," Microprocessors and Microsystems, vol. 62, pp. 91–101, 2018.

[11] J. Ding and S. Li, "A modular multiplier implemented with truncated multiplication," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 11, pp. 1713–1717, 2017.

[12] B. Kro¨ se, B. Krose, P. Van der Smagt, and P. Smagt, "An introduction to neural networks", 1993.

[13] S. Das, S. Ghosh, P. Dasgupta, and S. Sensarma, "A rule-based method for minimizing power dissipation by reducing switching activity of digital circuits", 2015.

[14] S. Purohit and M. Margala, "Investigating the impact of logic and circuit implementation on full adder performance," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 20, no. 7, pp. 1327–1331, 2011.

[15] R. Menon, S. Chennupati, N. K. Samala, D. Radhakrishnan, and B. A. Izadi, "Switching activity minimization in combinational logic design." In ESA/VLSI. Cite seer, 2004, pp. 47–53.

[16] C. McIvor, M. McLoone, and J. V. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques," IEE Proceedings- Computers and Digital Techniques, vol. 151, no. 6, pp. 402–408, 2004.

[17] A. F. Tenca and C¸ . K. Koc¸, "A scalable architecture for modular multiplication based on Montgomery's algorithm," IEEE Transactions on computers, vol. 52, no. 9, pp. 1215–1221, 2003.

[18] A. F. Tenca, G. Todorov, and C. K. Koc¸, "High-radix design of a scalable modular multiplier," in International Workshop on Cryptographic Hardware and Embedded Systems. Springer, 2001, pp. 185–201.

[19] I. Brzozowski and A. Kos, "Minimisation of power consumption in digital integrated circuits by reduction of switching activity," in Proceedings 25th EUROMICRO Conference. Informatics: Theory and Practice for the New Millennium, vol. 1. IEEE, 1999, pp. 376–380.

[20] P. Kornerup, “A systolic, linear-array multiplier for a class of right-shift algorithms”, IEEE Transactions on computers, vol. 43, no. 8, pp. 892–898, 1994.

[21] F. J. Taylor, “A VLSI residue arithmetic multiplier,” IEEE Transactions on Computers, vol. 31, no. 06, pp. 540–546, 1982.