Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130509

Adapted Semi Carry Save Modular Montgomery Multiplier for High-Performance Elliptic Curve Cryptographic Processor

S. Senthilkumar[1,*] *and P. S. Periasamy*[2]

[1] Department of Electrical and Electronics Engineering, K.S.R. College of Engineering, Tiruchengode, India
[2] Department of Electronics and Communication Engineering, K.S.R. College of Engineering, Tiruchengode, India

Received: 12 Jan. 2019, Revised: 12 Apr. 2019, Accepted: 13 Apr. 2019
Published online: 1 Sep. 2019

Abstract: In this paper,we propose a new adaptable Semi Carry Save Modular Montgomery Multiplier (ACS-MMM) with Multi-Value Logic method (MVL). We improve the performance of Modified Carry Save Adder (MCSA) with maintaining very small hardware complexity. The proposed ACS-MMM multiplier design has several advantages over previous models. The existing system requires additional clock cycles and the critical path delay parameter is reduced.It is shown that one can avoid the ACS Montgomery Modular Multiplier by avoiding unnecessary carry save addition measures that offer less delay. Therefore, the required clock cycles are significantly reduced by montgomery modular process. As a result, the proposed montgomery multiplier gives much smaller Area-Time Product (ATP) than the previous montgomery multipliers.

Keywords: Elliptic Curve Cryptography, Carry Save Adder, Area Time Product, Modular Multiplication and Power dissipation

1 Introduction

Cryptography is the science or art of securing data. It is the study of information hiding and verification. Cryptography has crept into everything, from web browsers and e-mail programs for cell phones, bank cards, cars and even into medical implants, voting system. A public key cryptography is needed for complex mathematical calculation in Elliptic Curve Cryptography (ECC) processor. The elliptic curve is used in cryptography applications because of its individual mathematical properties, it is applied to both encryption and decryption process. Neil Koblitz and Victor Miller in 1985 discussed elliptic curves in public key cryptography method [1, 2]. An elliptic curve is very specific and it makes the signal stronger with less number key size and it replaces RSA. In addition to the prime factor the RSA protocol provides much more protection when compared to ECC in the digital programmable logic [3–6].

A Residual Number System (RNS) is provided by an elliptical curve point multiplier architecture that is used in ECC processor [7–10]. Such an application is feasible and has significant improvement in the multiplication process. In [11–13], authors discussed the elliptic curve scalar multiplier hardware architecture of the $GF(2^m)$. A pseudo-word pipeline series, finite field multiplier and scalar multiplication is developed by [14–16] with the word size of w. Elliptic Curve Cryptography (ECC) with dual field processor is presented in [17, 18]. The scalar multiplication is further implemented in this structure. Both the prime and binary fields are used to implement ECC. It consists of a control unit with 256 bit serial and parallel operations. A new elliptic curve encoding processor structure is presented in [19–23]. The processor supports both field of operation (prime and binary field). In [24] Senthilkumar and Periasamy discussed four different multiplier designs such as DMM-Optimized Carry Look Ahead Adder (OCLA), DMM-Optimized Carry Bypass Adder (OCBA), DMM-Look up Table Carry Select Adder (LCSLA) and Dual Field Vedic Multiplier (DVM-LCSLA) are used to increase the performance of the ECC processor. The above discussed methodologies have some issues with a cryptography system, to overcome above problems new approach is introduced. The target of this research work is to overcome power dissipation and latency issues.

* Corresponding author e-mail: s.senthilkumarksrce@rediffmail.com

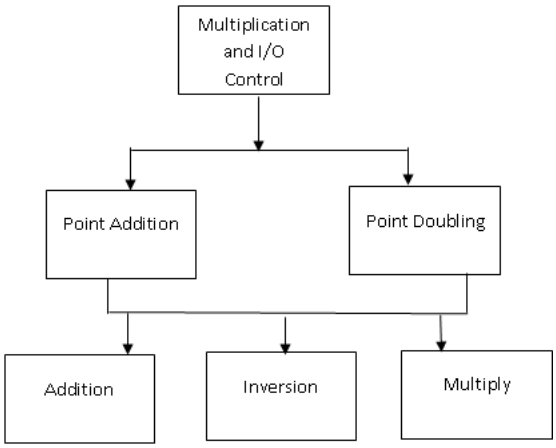

Fig. 1: Elliptic curve cryptosystem design hierarchy

2 Elliptic Curve Cryptosystem

Cryptography is essential for security purpose in all data transmission. The hierarchy of the elliptic curve cryptosystem is shown in Fig. 1. For its hardware implementation of multiplier the montgomery modular multiplier is used. The most important component in all multiplier unit is adder. The adder used in montgomery algorithm can be classified into two types based on its operation. They are Semi Carry Save Montgomery Modular Multiplication (SCS-MMM) and Full Carry Save Montgomery Modular Multiplication (FCS-MMM). In FCS-MMM both some and carry have been considered as an output. But in case of SCS-MMM only the sum is recognized as output. When compared to FCS-MMM, SCS-MMM has a low area because less number of adder is required.

In this work montgomery modular multiplier based on semi carry save with Multi Value Logic (MVL) algorithm is proposed. The MVL algorithm is one of the most well known in montgomery multiplier.

3 ECC Architecture

The proposed semi carry save montgomery multiplier based ECC architecture is shown in Fig. 2. The Multi Value Logic (MVL) algorithm-based Semi Carry Save Modular Montgomery Multiplier (SCS-MMM) is designed to reduce critical path delay. The ECC architecture has the following main components:

1. Register Bank
2. Control Unit
3. Adapted semi-carry save modular montgomery multiplier

Table 1: Utility functions registers in register bank

S. No	Register	Description
1	RA1	1. During Initialization, it is loaded with Px 2. Stores the X coordinate of the result 3. Also used for temporary storage
2	RA2	Store Px
3	RB1	1. During Initialization, it is loaded with Py 2 Stores the y coordinate of the result 3. Also used for temporary storage
4	RB2	Store Py
5	RB3	Used for temporary storage
6	RB4	Stores the constant curve b
7	RC1	1. During initialization, it is set to 1. 2. Store Z coordinate of the projective result
8	RC2	Used for temporary storage

3.1 Register Bank

The register file contains eight registers; the size of each register is 233 bits. Each clock rotation of register helps to save the results of calculations. Table 1 shows the utility functions of registers in register bank.

3.2 Control Unit

At every clock cycle, the control unit produces a control signal. The control word signals control the flow of data and also decide the operations performed on the data. The totally Thirty three control signal are generated by the control unit.

3.3 Semi Carry Save Montgomery Modular Multiplier with MVL

A new montgomery modular multiplier with MVL algorithm is used to overcome the drawbacks of existing multiplier design. In proposed multiplier quotient precomputation, transition time delay and number of clock cycle required are reduced. A semi carry save modular montgomery multiplier with multi value logic algorithm steps are given in Algorithm 1.

In this work one level CCSA adder architecture is introduced to decrease the required clock cycles. The steps for $\hat{Y}$ productivity and $\hat{D}$ are made from 1–8. Note that, this is due to creation in the ith iteration of the $Q_j + 1$ and $Q_j + 2$, the montgomery modular multiplication should be replaced by the $I-1$ index of the multiplication and the initial values must be set to 0 for $\hat{Q}$ and $\hat{X}$ respectively. In addition, the loop is replaced by proposed multiplier, if $Skip_{j+1} = 1$. The proposed multiplier avoids the unwanted iterations. Furthermore, MVL algorithm is replaced by $M+4$ instead of $M+1$. This is because of Y being superseded by $\hat{Y}$ and two-dimensional computing division thus making sure

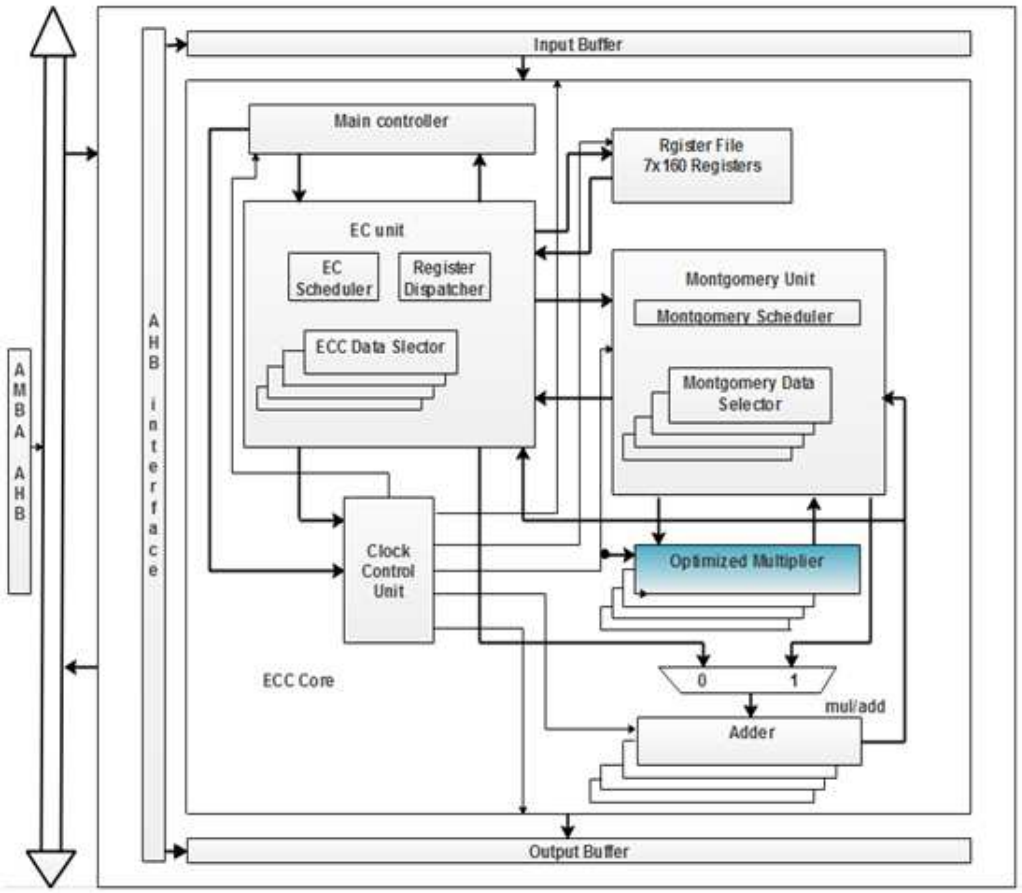

Fig. 2: Architecture of elliptic curve cryptography processor

that montgomery multiplier amplifies the three additional disadvantages. In while loop, Steps 11–12 are offered by a 4-to-1 multiplexer with a standardized CCSA architecture. In Step 12 the selection of j, Q and X are done and the calculations of $Skip_{j+1}Q_j + 1$ and $Q_j + 2$ are performed in Step 11 which can be run in parallel. Step 11 remembers that the next clock cycle may be delayed by reducing architectural transition delay.

3.4 The Hardware Architecture of SCS-MMM with MVL

The hardware architecture of semi carry save modular montgomery multiplier with MVL algorithm is shown in Fig. 3. The proposed multiplier consists of the following main blocks such as skip detector ($Skip_D$), six registers, zero detector ($Zero_D$), two sets of 4:1 multiplexers and one simplified multiplier SM3. The M4 and M5 are two multiplexers for 3-bit and these multiplexes are smaller than M1, M2. Moreover, the level of M-bit is less than the $Skip_D$.

Towards the beginning of the montgomery multiplier operation, $Skip_{j+1}$ is stored in FFs. $\hat{Q}$, and $\hat{X}$ are first reset to 0, then $\hat{D} = \hat{B} + \hat{N}$ design can be calculated through a CCSA.

The circuit diagram of skip detector is depicted in Fig. 4 and generate the output of $Skip_{j+1}$ and $\hat{Q}$. It comprises of four main blocks, such as a OR gate, 3 AND gates 4 XOR gates and two 2:1 multiplexers. It initially makes $Qi + 1$, $Qi + 2$, and j would prefer not to avoid the $j + 1$ flag of ith iteration from 5 to 8 stage individually and afterwards select the right $\hat{Q}$ and $\hat{X}$ indicated by $Skip_{i+1}$. While completing the ith iteration, $\hat{Q}$,$\hat{A}$, and $Skip_{i+1}$ must be put away to FFs.

In the ith iteration of the multiplier SM3 provides the proper output based on the $\hat{Q}$, $\hat{X}$ and also the output of sum & carry has been modified with the help of $M1$ and $M2$ based on $Skip_{i+1}$.

If the output of $Skip_{j+1}$ is equal to zero, then the carry value is shifted to one level ($Carry \gg 1$ and $Sum \gg 1$) otherwise it shifted to two level ($Carry \gg 2$ and $Sum \gg 2$). In addition, M4 and M5 can also be selected and the output is calculated in accordance with $Carry\ [j]2 : 0$

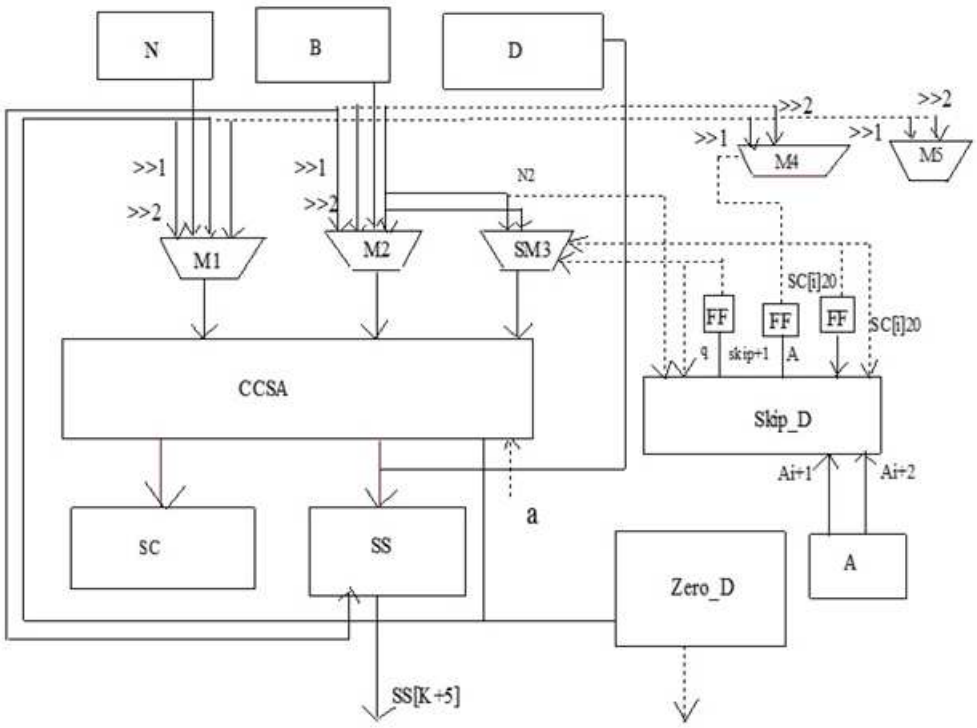

Fig. 3: ACS-MMM architecture

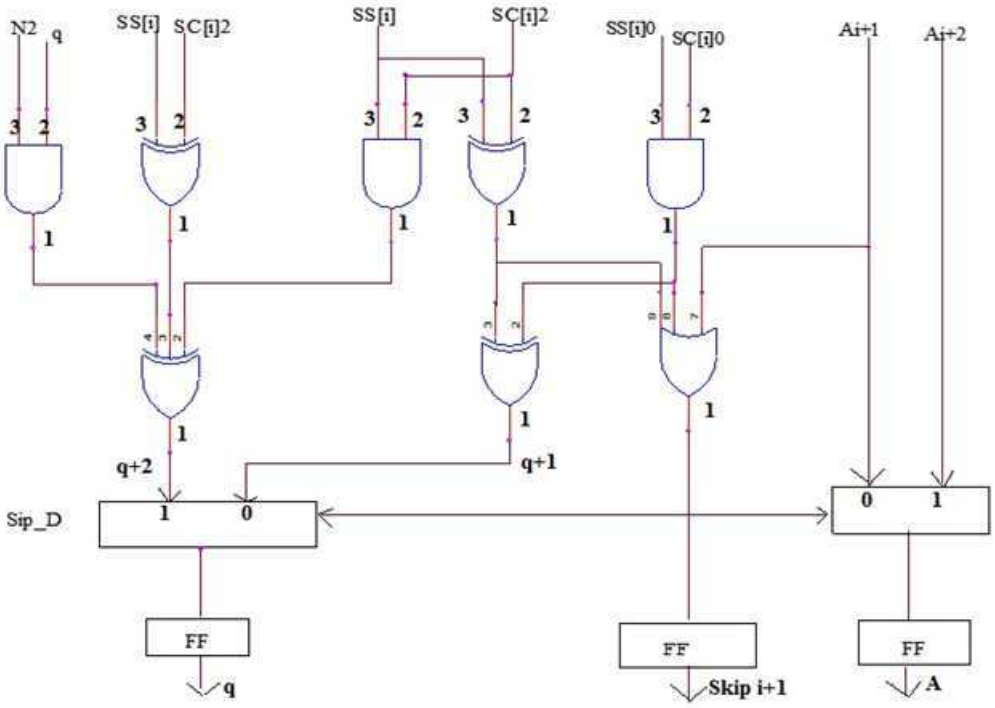

Fig. 4: Circuit diagram of skip detector

Algorithm 1: Semi Carry Save Montgomery Modular Multiplier with MVL Algorithm

Inputs : $X, Y, \hat{Z}$ (New Modules)
Outputs: Sum $[M+5]$
Step 1: $\hat{Y} = Y \ll 3$; $\hat{Q} = 0$; $\hat{X} = 0$ $Skip_{i+1} = 0$;
Step 2: $(sum, carry) = IF\_CSA(\hat{B}\hat{N}, 0)$
Step 3: While $(CARRY! = 0)$
$(sum, carry) = 2H\_CarrySaveAdder(Sum, Carry)$;
$\hat{B} = SS$;
$j = -1$; $Sum[-1] = 0$; $CARRY[-1] = 0$;
Step 4: While $(j \leqslant M+4)$
{
Step 5: If $(\hat{X} = 0$ and $\hat{Q} = 0)x = 0$;
Step 6: If $(\hat{X} = 0$ and $\hat{Q} = 1)x = \hat{Z}$;
Step 7: If $(\hat{X} = 1$ and $\hat{Q} = 0)x = \hat{Y}$;
Step 8: If $(\hat{X} = 1$ and $\hat{Q} = 1)x = \hat{D}$;
Step 9: $(Sum[j+1], Carry[j+1]) = IF\_CarrySaveAdder(Sum[j], Carry[j], x) \gg 1$;
Step 10: Compute $Q_{j+1}Q_{j+2}$ and $Skip_{j+1}$
Step 11: If $(Skip_{j+1} = 1)$
{
$Sum[j+2] = Sum[j+1] \gg 1$;
$Carry[j+2] = Carry[j+1] \gg 1$;
$\hat{Q} = Q_{j+2}$; $\hat{X} = X_{j+2}$; $i = i+2$}
Step 12: Else
{
$\hat{Q} = Q_{j+1}$; $\hat{X} = X_{j+1}$; $i = i+1$}
}
$\hat{Q} = 0$; $\hat{X} = 0$
Step 13: While $(Carry[M+5]! = 0)$
$(Sum[M+5], Carry[M+5]) = 2H\_CarrySaveAdder(Sum[M+5], Carry[M+5])$
Step 14: Return $Sum[M+5]$;

and $Sum\ [i]2:0$ to generate $Skip_{i+1}$ in ith iteration. Note that the multiplexers M1 and M1 give the output of $Carry\ [i]2:0$ and $Sum\ [i]2:0$. In this multiplexers take longer delay because they are $4:1$ multiplexers. After completing the while loop, $\hat{Q}$ and $\hat{X}$ stored in FFs are reset to 0. At that point, the multiplication in stages 12 and 14 can be performed by the semi carry save montgomery modular multiplier with MVL. The calculation of $\hat{D} = \hat{B} + \hat{N}$ is caluculated from the Steps 3 to 4. Finally, $Sum\ [M+5]$ in binary format is calculated and $Carry\ [M+5]$ is equal to 0.

4 Results and Discussion

The proposed SCS-MMM with MVL-based elliptic curve cryptographic processor is designed and implementation has been done using Xilinx ISE Design Suite 14.1 in FPGA Virtex-6 device. From the results, the proposed montgomery multiplier design gives very small Area Time Product (ATP) when compared to the previous montgomery multiplier. As compared to other multipliers, the proposed multiplier has a low area because of less number of adder levels in the underlying algorithm.

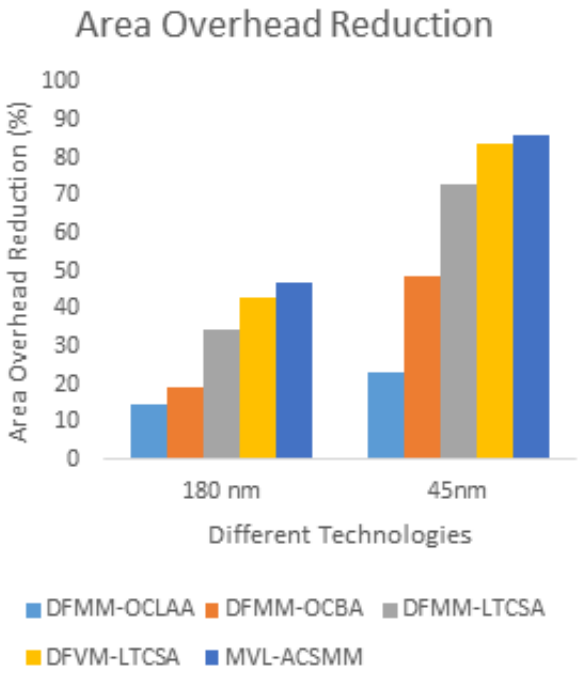

Fig. 5: Area overhead reduction analysis

Table 2: Area overhead analysis for different methods

S. No	Methods	Technology	Hardware Area Overhead reduction (%)
1	DFMM-OCLAA	180 nm	14.78
		45 nm	22.94
2	DFMM-OCBA	180 nm	19.41
		45 nm	48.5
3	DFMM-LTCSA	180 nm	34.72
		45 nm	73.1
4	DVM-LTCSA	180 nm	42.77
		45 nm	84.01
5	SCS MMM MVL	180 nm	46.77
		45 nm	86.01

Table 2 and Fig. 5 display the hardware area overhead reduction analysis with a different of multiplier architectures. This architecture is implemented in both 180 nm and 45 nm technologies. As compared with existing multiplier design the proposed multiplier (SCSMMM-MVL) gives better result for both 180 nm and 45 nm. The area overhead reduction are (46.77%) and (86.01%) for 180 nm and 45 nm technologies respectively.

Table 3: Power consumption reduction analysis

S. No	Methods	Technology	Power Consumption Reduction
1	DFMM-OCLAA	180 nm	10.15
		45 nm	29.73
2	DFMM-OCBA	180 nm	11.43
		45 nm	38.39
3	DFMM-LTCSA	180 nm	41.60
		45 nm	54.72
4	DFVM-LTCSA	180 nm	71.09
		45 nm	72.79
5	SCS MMM MVL	180 nm	74.63
		45 nm	78.42

Table 4: Time delay reduction analysis

S. No	Methods	Technology	Time Delay reduction (%)
1	DFMM-OCLAA	180 nm	13.38
		45 nm	5.30
2	DFMM-OCBA	180 nm	19.50
		45 nm	18.20
3	DFMM-LTCSA	180 nm	27.25
		45 nm	25.16
4	DFVM-LTCSA	180 nm	28.61
		45 nm	20.33
5	SCS MMM MVL	180 nm	29.61
		45 nm	21.33

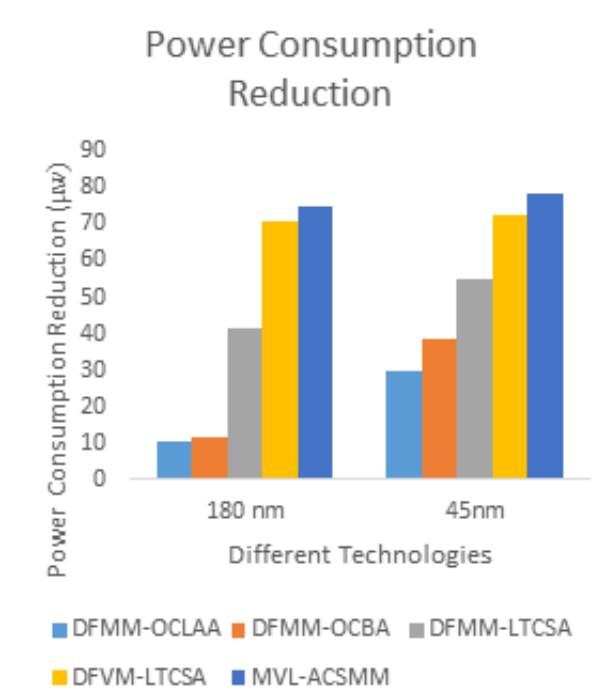

Fig. 6: Power consumption reduction analysis

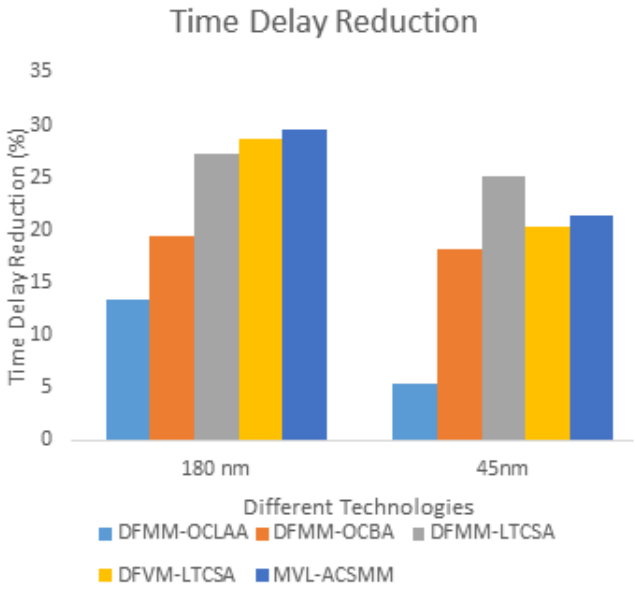

Fig. 7: Time delay reduction analysis

Table 3 and Fig. 6 present the power consumption reduction (%) analysis with different types of multiplier architectures. This architecture's result has been taken in both 180 nm and 45 nm technologies. As compared with previous design, the proposed adaptive Semi Carry Save Montgomery Modular Multiplier with Multi Value Logic (SCSMMM-MVL) delivers the better results for both 180 nm (74.63) and 45 nm (78.42) technologies.

Table 4 and Fig. 7 present the reduction (%) of critical path time delay with different types of multiplier architectures. This architecture result has been taken in both 180 nm and 45 nm technologies. As compared with existing design the proposed Semi Carry Save Montgomery Modular Multiplier with Multi Value Logic (SCSMMM-MVL) delivers the better results for both 180 nm (29.61%) and 45 nm (21.33%) technologies.

5 Conclusion

In this paper, semi carry save montgomery modular multiplier architecture for cryptography system using multi value logic method is introduced. The proposed SCS-MMM with MVL is designed and implemented in Xilinx by using Verilog code. In this multiplier, instead of using usual accumulator, the LCSLA accumulator has been used to evaluate the concert constraints like area, power, and time delay. The SCS-MMM with MVL

method has given better results in both FPGA and ASIC. From the FPGA implementation results the numbers of LUT, slices, flip-flops, and frequency have been improved in SCS MMM with MVL method when compared to existing. Hence the hardware area overhead reduction of (86.01%), power reduction of (74.63%) and reduction in time delay of (29.61%) have been achieved in the proposed SCS-MMM with MVL in 180 nm technology and similarly the hardware area overhead reduction of (46.77%) power reduction of (78.42%) and time delay reduction of (21.23%) are achieved in 45 nm technology.

References

[1] R.L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *ACM Communication*, **21**(2), 120–126 (1978).

[2] Som, Subhranil, Majumder, Rana and Dutta, Elliptic curve cryptography: A dynamic paradigm, 10.1109/ICTUS.2017.8286045, 427–431 (2017).

[3] N. Koblitz, Elliptic Curve Crypto systems, *Math. Comput.*, **48**(177), 203–209 (1987).

[4] Kedarisetti, Karthik, Gamini, Roopesh and V. Thanikaiselvan, Elliptical Curve Cryptography for Images Using Fractal Based Multiple Key, Hill Cipher, 10.1109/ICECA.2018.8474689, 643–649 (2018).

[5] Y.S. Kim, W.S. Kang and J.R. Choi, Asynchronous implementation of a 1024-bit modular processor for RSA cryptosystem, in *Proc. 2nd IEEE Asia-Pacific Conf. ASIC*, 187–190 (2000).

[6] V. Bunimov, M. Schimmler and B. Tolg, A complexity-effective version of Montgomery's algorithm, in *Proc. Workshop Complex. Effective Designs* (2002).

[7] H. Zhengbing, R.M. Al Shboul and V.P. Shirochin, An efficient architecture of 1024-bits crypto processor for RSA cryptosystem based on modified Montgomery's algorithm, in *Proc. 4th IEEE Int. Workshop Intell. Data Acquisition Adv. Comput. Syst.*, 643–646 (2007).

[8] Y.-Y. Zhang, Z. Li, L. Yang and S.-W. Zhang, An efficient CSA architecture for Montgomery modular multiplication, *Microprocessors Micro-system*, **31**(7), 456–459, (2007).

[9] C. McIvor, M. McLoone and J.V. McCanny, Modified Montgomery modular multiplication and RSA exponentiation techniques, *IEEE Proc.-Comput. Digit. Techn.*, **151**(6), 402–408 (2004).

[10] S.-R. Kuang, J.-P. Wang, K.-C. Chang and H.-W. Hsu, Energy-efficient high throughput Montgomery modular multipliers for RSA cryptosystems, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, **21**(11), 1999–2009 (2013).

[11] J.C. Neto, A.F. Tenca and W.V. Ruggiero, A parallel k-partition method to perform Montgomery multiplication, in *Proc. IEEE Int. Conf. Appl. Specific Syst., Archit., Processors*, 251–254 (2011).

[12] J. Han, S. Wang, W. Huang, Z. Yu and X. Zeng, Parallelization of radix-2 Montgomery multiplication on a multicore platform, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, **21**(12), 2325–2330 (2013).

[13] Ranbir Singh, Soram, Kumar Khan, Ajoy, Sonamani Singh and Takhellambam, A critical review on Elliptic Curve Cryptography, 10.1109/ICACDOT.2016.7877543, 13–18 (2016).

[14] G. Sassaw, C.J. Jimenez and M. Valencia, High radix implementation of Montgomery multipliers with CSA, in *Proc. Int. Conf. Microelectron*, 315–318 (2010).

[15] A. Miyamoto, N. Homma, T. Aoki and A. Satoh, Systematic design of RSA processors based on high-radix Montgomery multipliers, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, **19**(7), 1136–1146 (2011).

[16] S.-H. Wang, W.-C. Lin, J.-H. Ye and M.-D. Shieh, Fast, scalable radix-4 Montgomery modular multiplier, in *Proc. IEEE Int. Symp. Circuits Syst.*, 3049–3052 (2012).

[17] Azarderakhsh, Reza, and Arash Reyhani Masoleh, Parallel and high-speed computations of elliptic curve cryptography using hybrid-double multipliers, *IEEE Transactions on Parallel and Distributed Systems*, **12**(4), 1668–1677(2015).

[18] He Debiao, Huaqun Wang, Muhammad Khurram Khan and Lina Wang, Lightweight anonymous key distribution scheme for the smart grid using elliptic curve cryptography, *IET Communications*, **10**(14), 1795–1802 (2016).

[19] Yeh, Hsiu-Lien, Tien-Ho Chen, Kuei-Jung Hu and Wei-Kuan Shih, Robust elliptic curve cryptography-based three factor user authentication providing privacy of biometric data, *IET Information Security*, **7**(3), 247–252 (2013).

[20] He Debiao and Sherali Zeadally, An analysis of RIFD authentication schemes for an internet of things in healthcare environment using elliptic curve cryptography, *IEEE Internet of Things Journal*, **2**(1), 72–83 (2015).

[21] Kimmo U. Jarvinen and Mehran Mozaffari Kermani, Efficient algorithm and architecture for elliptic curve cryptography for extremely constrained secure applications, *IEEE Transactions on Circuits and Systems I*, **5**(6), 1144–1155 (2014).

[22] Kuang, Shiann-Rong, Kun-Yi Wu and Ren-Yao Lu, Low-cost, high-performance VLSI architecture for Montgomery modular multiplication, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **5**(3), 434–443 (2016).

[23] Yan Zhao, Shiming Li and Liehui Jiang, Secure and Efficient User Authentication Scheme Based on Password and Smart Card for Multiserver Environment, Security and Communication Networks, https://doi.org/10.1155/2018/9178941 (2018).

[24] S. Senthilkumar and P.S. Periasamy, Power Efficient Implementation of ECC Using LCSLA Based Dual Field Vedic Multiplier, *Journal of Measurement and Control* (2017).

S. Senthilkuamar obtained his B.E. degree in Electrical &Electronics Engineering from Adhiparasakthi Engineering College,Melmaruvathur in 2006. He obtained his M.E. degree in Applied Electronics from Kongu Engineering College, Perundurai in 2008 and Pursuing his Ph.D. degree in Anna University, Chennai. He is working as Assistant Professor in K.S.R. College of Engineering, Tiruchengode, Tamil Nadu,India. He has published research papers in International conferences and Journals. His areas of interest are: Low Power VLSI System, Cryptography and Network Security.

P.S. Periasamy was born in Tiruchengode, Tamilnadu on 22nd July1972. He obtained his B.E. degree in Electrical &Electronics Engineering from Government College of Engineering, Salem in 1998. He obtained his M.E. degree in Electronics and communication engineering from Government College of Technology, Coimbatore in 2003 and his Ph.D. degree from Anna University, Chennai in 2011. He worked as a lecturer in department of EEE in K.S. Rangasamy institute of technology from 1998 to 2001. Then he worked as an Assistant professor in department of ECE from 2001 to 2004 and an Associate professor in department of ECE from 2004 to2006. After that he was promoted as Professor and Head in 2006 and he continues as Professor& Head /ECE till now in K.S.R. College of Engineering, Tiruchengode, TamilNadu, India. He has published several research papers in International and National Journals. He has also presented research papers in National and International conferences. His areas of interest are: Digital Signal & Image Processing, Wireless and Computer Networks.