International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

# Area and Delay Efficient Compressor Based Montgomery Multiplier

**G. Revathi, K. V. Gowreesrinivas, P. Samundiswary**

***Abstract***: ***Modular multiplication plays an important role in public-key crypto-systems. It first performs integer multiplication and then the costlier division operation is done. Due to this, the computation time and resource requirements will be high. In order to overcome these limitations, Montgomery Modular Multiplication (MMM) method is widely used. As integer multiplication is the most important operation in the Montgomery Multiplication algorithm, it becomes mandatory to enhance the speed of Integer Multiplier (IM) so that it can increase the overall efficiency of Montgomery Multiplier (MM). This work is mainly focused on compressor based MM to improve the performance in respect to propagation delay and area. In this paper, MM using 3:2, 4:2 and 5:2 compressors based Array Multiplier (AM) are designed for performing Integer Multiplication in MM and their performance are evaluated. The modules are synthesized using Xilinx ISE 14.7 and targeted on FPGA family Artix-7. Finally, comparative analysis is made in terms of delay and area.***

***Keywords: Compressors, ISE, Modular multiplication, Montgomery***

## I. INTRODUCTION

Modular arithmetic is an arithmetic system in which the remainder is considered. In this system, numbers depend upon a certain value called modulus to leave the remainder. It is often operated on prime numbers. Basic operations in modular arithmetic include addition, multiplication, division, exponentiation and multiplicative inverses. Among this, modular multiplication is widely used because the time and resource consumption is more than modular addition and subtraction. It is widely used in cryptographic applications. Cryptography is the study and practice of secure communication techniques. It deals with protocols for preventing third parties or the public from reading private messages. It is mainly divided into Asymmetric Key (AK) and Symmetric Key (SK) cryptography. AK cryptography is also named as Public Key Cryptography (PKC). It has a public key and a private key. The public key is used for data encryption and the corresponding private key is used for data decryption. Rivest, Shamir, Adleman (RSA) and Elliptic Curve Cryptography (ECC) schemes are commonly used in PKC algorithms. PKC schemes need modular arithmetic operations with huge operands. As the operand size increases, the protection is also increased.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

**G.Revathi***, Department of Electronics Engineering, Pondicherry University, Puducherry, India.

**K.V.Gowreesrinivas**, Department of Electronics Engineering, Pondicherry University, Puducherry, India.

**P.Samundiswary**, Department of Electronics Engineering, Pondicherry University, Puducherry, India.



When compared to RSA, ECC needs smaller operands for same level of protection. So, many PKC schemes are implemented using ECC. It uses modular multipliers so efficiently which results in the high speed of operation. Its algorithms are so flexible but they are too time consuming. Therefore, modular multipliers should be optimized in order to increase its perfosrmance. Modular multipliers are classified into three types [1]. The first type is Division over a modulus M. It is a standard method for modular multiplication computation. The drawback is that it is so costly because the time and resource requirements are high. The second method is Interleaved Modular Multiplication (IMM). In this method, the final result is obtained by reduced intermediate results. The final method is the best collective method for performing modular multiplication which is called as Montgomery Modular Multiplication (MMM). Even though both IMM and MMM methods are flexible, the Montgomery method is faster compared to that of the Interleaved method for the huge operands. Hence, in this paper an attempt has been made to develop MM using different compressors based multipliers Compressors acts as the essential building block which is used to speed up the multiplication process taking place by accumulating the partial products generated. Basically in DSP applications, the need of compressor based addition is more to improve the performance in terms of speed and power. It reduces the partial products and thereby reduces the critical path and hence the performance of the circuit is increased. There are various kinds of compressors available such as 3:2, 4:2, 5:2, and 5:3 and so on. The rest of the portion is organized as follows: In section II the related work to Montgomery multiplier and compressors are discussed. The architecture of MM and compressors based architecture are explained in Section III and Section IV. The proposed work is explained in Section V followed by the results discussion in Section VI. The performance comparison is shown in Section VII and Section VIII the Conclusion.

## II. RELATED WORK

Several ideas are proposed for the efficient MMM hardware implementation.

Four different designs of approximate 15:4 and 5:3 compressor are described in 16 bit multiplier by T.U.S Krishna [2]. Pixel by pixel multiplication of two images is done using approximate computing so that the complexity and power consumption is reduced. A multiplier circuit is designed by T.Adiono [3] using only primitive gates, adders, shifters, multiplexers and registers to execute Montgomery multiplication so as to use in asymmetric RSA cryptosystem.

The algorithm used is adaptable so that it can be reconfigured for applications with any arbitrary bits. S.Venkatachalam [4] proposed two variants of 16 bit multipliers approximation with better precision and power. The Montgomery multiplier with various bits size is comparatively described by C.Anoop [5] and its different parameters are categorically analyzed. The delay and area of Montgomery multiplier are increased as the bit size is increased. The 8192 bit Montgomery multiplier is designed by Y.Mo [6] based on Cox-Rower architecture. The redundancy, area and delay get reduced considerably.

A Configurable CSA (CCSA) is proposed by S.R.Kuang [7] and a mechanism is developed for maintaining the minimum critical path delay. The complete performance analysis of the Modular Multiplication using Montgomery and Interleaved techniques are described by K.Javeed [8]. Faster Montgomery algorithm for performing fast modular multiplication is proposed by N.Thampi [9]. It has a simple structure which needs few prior computation and storage.

A newly modified design of Montgomery multiplier algorithm is presented by A.Thomas [10] for better efficiency. 2 hardware architectures that compute Montgomery multiplication iteratively is proposed which is described by M.Morales [11]. G.S.Lakshmi [12] designed a 4:2 Compressor based 8 bit Vedic Multiplier using reversible logic and is compared with conventional multipliers.

Montgomery multiplier for filtering application is developed by N.Mulla [13]. A hybrid multiplier with 4-level recursion is constructed by X.Yan [14] by applying the Knuth and Karatsuba algorithm recursively. The double Booth–encodings and precomputation methods is applied to radix-4 Montgomery multiplier by T.Wu [15] which is scalable and feed forward. A fair comparison is provided by S.Kakde [16] and it includes the improved Montgomery multiplier design with CSA stages and parallel register.

4: 2 compressor is proposed by A. Pishvaie [17] using decomposition of XOR/XNOR gates. For evaluating each design's performance, nearly 1300 (4:2) compressors are incorporated in 54×54-bit binary multipliers. A. Pishvaie [18] designed three new 4:2 compressors which is constructed using optimized CMOS full-adder. In order to explain the functionality, another two 4:2 compressor are proposed.

A Montgomery multiplier incorporated with Karatsuba algorithm is designed by G.C.Chow [19].A 256-bit Montgomery modular multiplier using Karatsuba - Ofman algorithm is designed by Y.Gong [20]. A pipeline structure is designed to achieve a higher throughput rate. For adding the accumulated partial products, ternary adders and generalized Parallel Compressors are used in the design by S.Gao [21]. Multipliers of different sizes were implemented and compared to the standard multipliers.

## III. MONTGOMERY MULTIPLIER

Modular multiplication is the elementary action in large number of public-key crypto-systems and there are many algorithms used for that system. In MM, the Integer Multiplier (IM) is responsible for overall efficiency. The algorithm and the architecture are explained below.

### *A. Montgomery Algorithm*

For multiplication of positive integers X and Y, this algorithm needs to calculate (X * Y) mod M, where M is a large prime. Normal division method is so costly to find the remainder. Montgomery multiplication performs simple shift and add operations instead of costly division operation. Here, the computation takes place after transforming the numbers into Residue Number System (RNS) domain and retransforming after computation. Radix R must be selected such that it should be greater than the value of modulus M and twice the power of bit length n. For this algorithm, R and M must be coprime. The modular multiplication result Z is $Z=X\times Y \bmod M$ where $0 < X,Y < M$. Algorithm 1[1] describes the Montgomery multiplication.

Montgomery Modular Multiplication algorithm is explained below [1]:

Input: X, Y, M; $n = \log_2 M$, $R = 2^n$ , $M1 = -M^{-1} \bmod R$

Output : $Z = X \times Y \times R^{-1} \bmod M$

i. $D \leftarrow X \times Y$
ii. $E \leftarrow D \times M1 \bmod R$
iii. $Z \leftarrow (D + E \times M)/R$
iv. If $Z > M$ then return $Z - M$
v. Else return Z

### *B. Architecture*

The architecture of the Montgomery multiplier consists of integer multiplier, register file, controller, adder, subtractor and a multiplexer. The overall architecture of the Montgomery multiplier according to [1] is shown below.

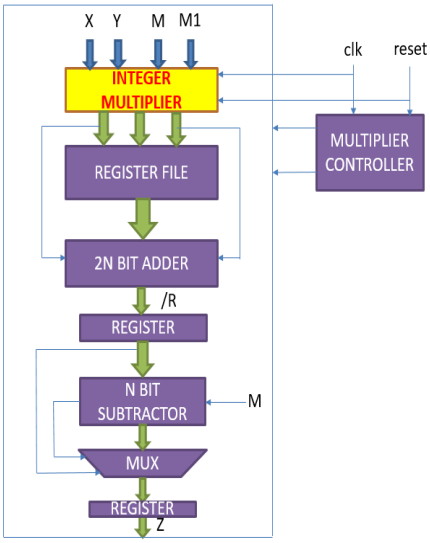



**Fig.1.MM Architecture [1]**

The efficiencys of the Integer multiplier influences the overall performance of Montgomery multiplier. In IM, three integer multiplications are computed in order and corresponding results are stored in registers. The first and last multiplication results are added. The final reduction gives the Montgomery result. The Montgomery multiplication is performed as described in the following steps according to the Algorithm 1 [1]. Firstly, the input registers X and Y get loaded and then the first integer multiplication takes place.

These results are stored in R1.Then again the input registers get loaded and now the second integer multiplication is performed between M1 and the modulus value of result stored in R1 and these results are stored in R2 registser. Now the input registers are loaded for performing final integer multiplication between M and the modulus value of result stored in R2.The multiplication results are added and the final result is compared accordingly and if required, subtraction takes place finally to obtain the result.

## IV. COMPRESSOR ARCHITECTURES

In this section, the architecture of 3:2, 4:2 and 5:2 compressors along with their internal diagram is explained.

### *A. 3:2 Compressor*

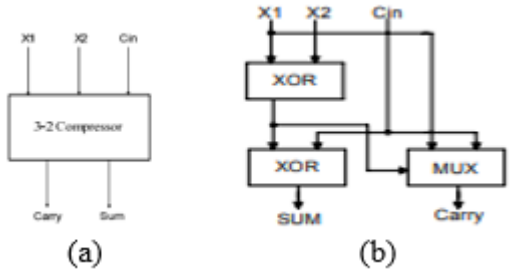



**Fig.2.Structure of 3:2 compressor (a) Symbol (b).Internal Diagram [22]**

The symbol and internal diagram of 3:2 compressor is shown in Fig.2.a and Fig.2.b.In 3:2 compressor, there are3 inputs and 2 outputs. It acts same as the full adder. The third input can also be $C_{in}$ if there is carry propagating from previous stage. The basic equation of this compressor is X1 + X2 + X3 = Sum + 2*Carry. [22]

### *B. 4:2 Compressor*

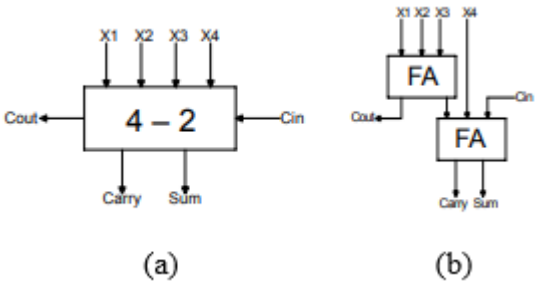



**Fig.3. Structure of 4:2 compressor (a) Symbol (b).Internal Diagram [22]**

The symbol and internal diagram of 4:2 compressor is illustrated in Fig.3.a and Fig.3.b.In 4:2compressor, there are 4 inputs and 2 outputs along with one $C_{in}$. The internal diagram of 4:2 compressor contains two full adders The basic equation of this compressor is X1 + X2 + X3 +X4 = Sum + 2*(Carry + Cout).[22]

### *C. 5:2 Compressor*

Fig.4.a and Fig.4.b illustrates the symbol and internal diagram of 5:2 compressor. In 5:2 compressor there are 5 inputs and 2 outputs along with two $C_{in}$. The internal diagram of 5:2 compressor contains three full adders. The basic equation of this compressor X1+X2+X3+X4+X5+Cin1+Cin2 =Sum+2*(Carry + Cout1 + Cout2). [22]

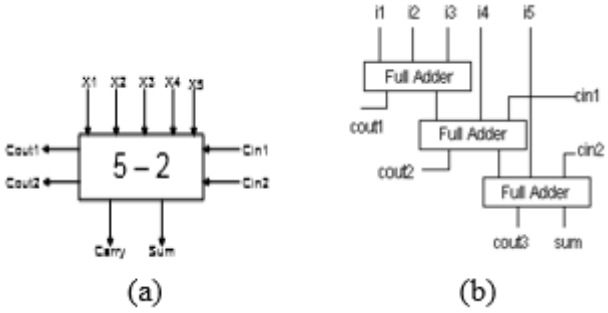



**Fig.4. Structure of 5:2 compressor (a) Symbol (b) Internal Diagram [22]**

## V. PROPOSED WORK

Integer multiplier present in Montgomery multiplier is wholly responsible for the overall efficiency of the Montgomery multiplication. Therefore, it becomes more important to improve the performance of IM in order to increase the efficiency of MM. Hence in this work, compressor based array multiplier is used for integer multiplication in MM which is shown in Fig.5

Array multiplier acts similar to the conventional type of multiplication. Because of its simple and regular structure it is widely used .In this, AND gates are used in generating Partial Products (PP) and those PPs gets added using half and full adders. In general, n-bit array multiplier utilizes n *(n-1) adders and $n^2$ AND gates.

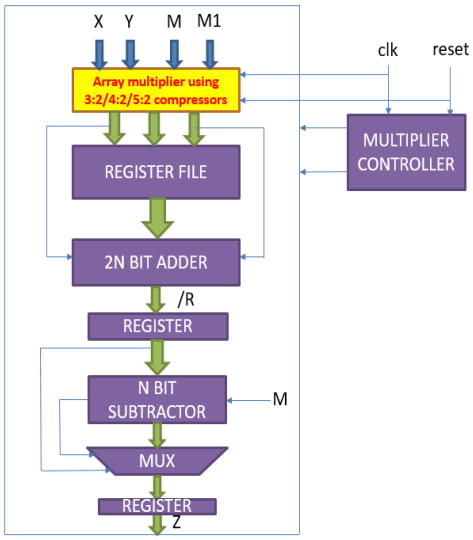



**Fig.5.MM using 3:2/4:2/5:2 compressor based AM**

Here the performance of MM is evaluated using 3:2, 4:2 and 5:2 compressors based array multiplier and compared to find out the best compressor based array multiplier. These are explained in three cases as follows.

*Case 1: MM using 3:2 compressor based Array multiplier*

In case 1, the array multiplier using 3:2 compressors is used as the integer multiplier in order to perform Montgomery multiplication.

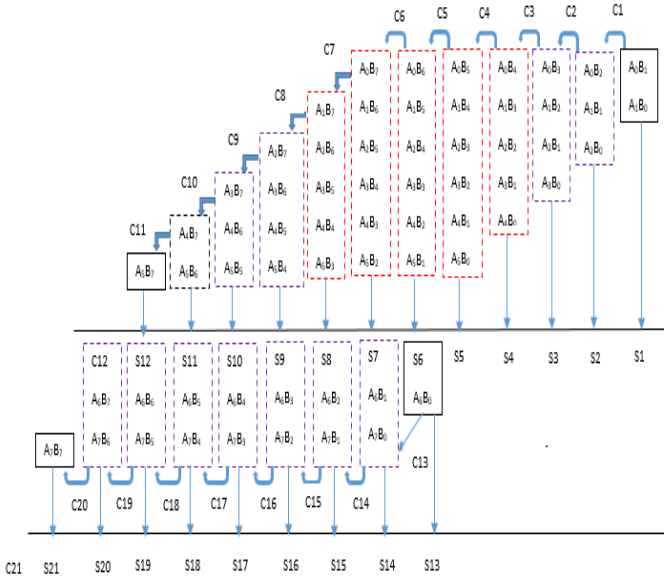
Here as demonstrated in Fig.5, the three integer multiplications takes place using 3:2 compressors based AM according to the Algorithm 1.The PPs gets generated using AND gates and those PPs gets added using 3:2 compressor adders.

*Case 2: MM using 4:2 compressor based Array multiplier*

In this case, the three integer multiplications takes place using 4:2 compressors based AM according to the Algorithm 1 as illustrated in Fig.5.The PPs are generated using AND gates and those PPs are added using 4:2 compressor adders.

*Case 3: MM using 5:2 compressor based Array multiplier*

In this case, the three integer multiplications takes place using 5:2 compressors based AM according to the Algorithm 1 as shown in Fig.5.The PPs gets generated using AND gates and those PPs gets added using 5:2 compressor adders. The block diagram of compressor based 8 bit array multiplication is shown below as the example to know how the compressors are being employed.



**.Fig.6. 5:2 compressors based 8 bit Array multiplication**

Fig.6 illustrates the steps included in 8 bit array multiplication using 5:2 compressors. This multiplication comprises both 4:2 as well as 3:2 compressors in it. The red and purple colour dotted box denotes 5:2 and 4:2 compressors. The black dotted box denotes 3:2 compressors. The partial products gets added up and the sum is represented as S1, S2 and so on. The carry gets propagated to the next step which is denoted as C1, C2 and so on.There will be totally two stages taking place. In first stage the addition starts using 3:2 compressor followed by two 4:2 compressors ,five 5:2 compressors then again continued by two 4:2 compressors and finally one 3:2 compressor and a half adder. The second stage contains only 4:2 compressors except first and last step which is using half adder.

## VI. RESULTS AND DISCUSSION

All the modules used in this work are coded with Verilog HDL and synthesized using Xilinx ISE 14.7. The Montgomery multiplication using compressors is synthesized and implemented and targeted on Artix-7 with device 7a100tcsg324-3.The RTL schematic, Area and Delay of 8 and 16 bit Montgomery multiplier using different compressors based array multiplier is shown in this section.

*A. RTL Diagram*

The RTL schematic of 8 and 16 bit MM using 3:2, 4:2 and 5:2 compressors based AM are shown below.

*i. RTL of 3:2 compressors based MM*

Here, the RTL view of 8 and 16 bit MM using 3:2 compressors based MM is shown in Fig.7 and Fig.8.

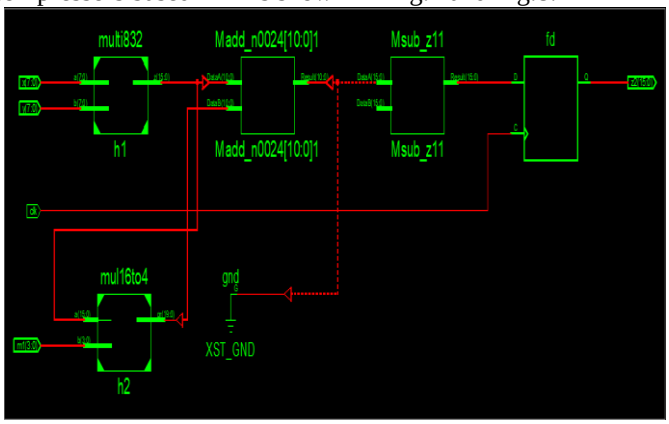



**Fig.7. RTL view of 8 bit MM using 3:2 compressor based AM**

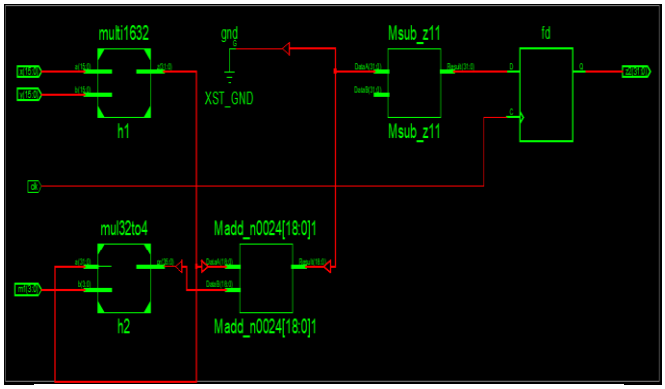



**Fig.8. RTL view of 16 bit MM using 3:2 compressor based AM**

*ii. RTL of 4:2 compressors based MM*

Here, the RTL view of 8 and 16

bit MM using 4:2 compressors based MM is illustrated in Fig.9 and Fig.10.

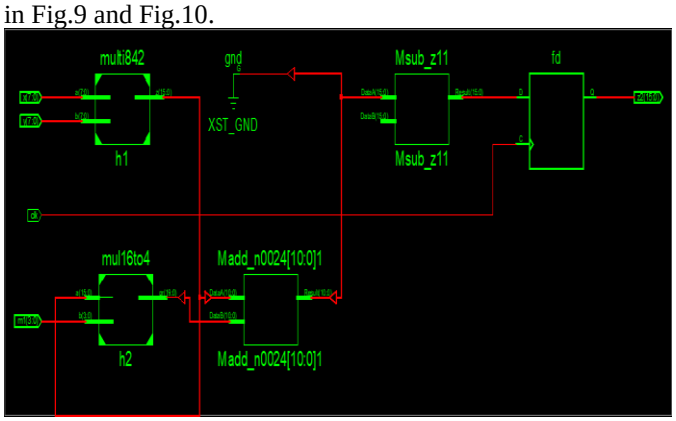



**Fig.9. RTL view of 16 bit MM using 4:2 compressor based AM**

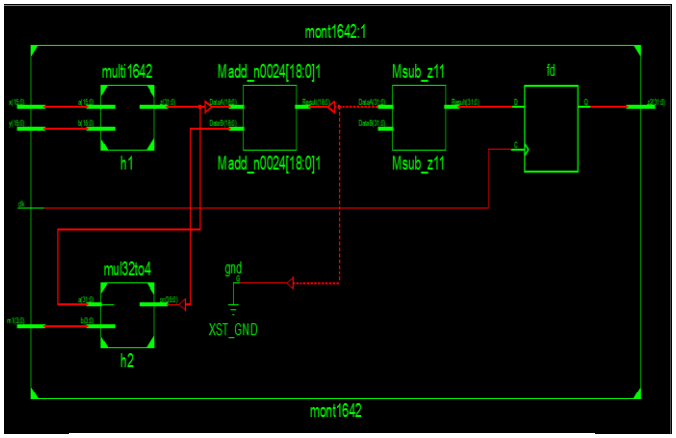



**Fig.10. RTL schematic of 16 bit MM using 4:2 compressor based AM**

*iii. RTL of 5:2 compressors based MM*

Here, the RTL view of 8 and 16 bit MM using 5:2 compressors based MM is shown in Fig.11 and Fig.12.

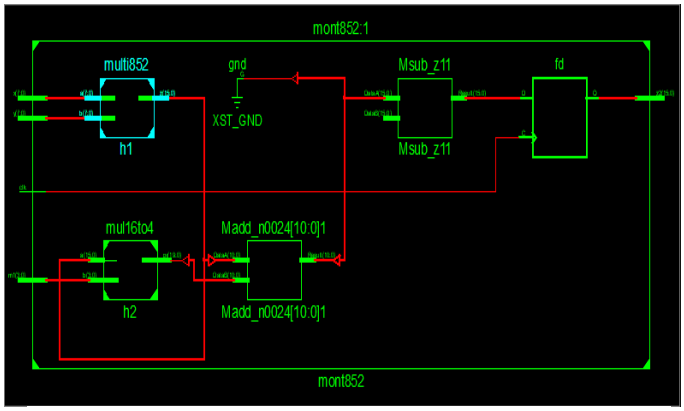



**Fig.11.RTL schematic of 8 bit MM using 5:2 compressor based AM**

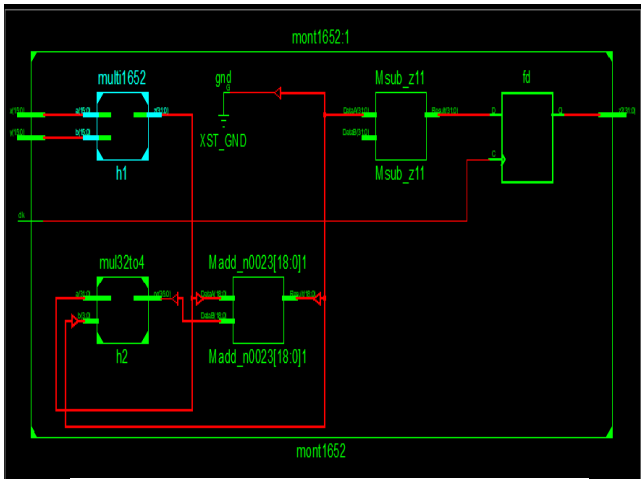



**Fig.12.RTL schematic of 16 bit MM using 5:2 compressor based AM**

*B. Area Analysis*

The area utilization of 8 and 16 bit MM using 3:2,,4:2 and 5:2 compressor based AM is explained below.

*i. Area of 3:2 compressors based MM*

Here, the area utilization of 8 and 16 bit MM using 3:2 compressor based MM is shown in Fig.13 and Fig.14.

| Device Utilization Summary | | |
|---|---|---|
| **Slice Logic Utilization** | **Used** | **Available** |
| Number of Slice Registers | 0 | 126,800 |
| Number of Slice LUTs | 91 | 63,400 |
| Number used as logic | 91 | 63,400 |
| Number using O6 output only | 65 | |
| Number using O5 output only | 0 | |
| Number using O5 and O6 | 26 | |
| Number used as ROM | 0 | |
| Number used as Memory | 0 | 19,000 |
| Number used exclusively as route-thrus | 0 | |
| Number of occupied Slices | 34 | 15,850 |

**Fig.13. Area utilization of 8 bit MM using 3:2 compressor based AM**

| Device Utilization Summary | | |
|---|---|---|
| **Slice Logic Utilization** | **Used** | **Available** |
| Number of Slice Registers | 3 | 126,800 |
| Number used as Flip Flops | 0 | |
| Number used as Latches | 0 | |
| Number used as Latch-thrus | 0 | |
| Number used as AND/OR logics | 3 | |
| Number of Slice LUTs | 375 | 63,400 |
| Number used as logic | 375 | 63,400 |
| Number using O6 output only | 318 | |
| Number using O5 output only | 0 | |
| Number using O5 and O6 | 57 | |
| Number used as ROM | 0 | |
| Number used as Memory | 0 | 19,000 |
| Number used exclusively as route-thrus | 0 | |
| Number of occupied Slices | 179 | 15,850 |

**Fig.14. Area utilization of 16 bit MM using 3:2 compressor based AM**

From Fig.13, it is observed that for 8 bit MM, it consists of 91 LUTs with 34 slices occupied. Similarly for 16 bit MM, it includes 375 LUTs with 179 slices occupied as shown in Fig.14.

*ii. Area of 4:2 compressors based MM*

Here, the area utilization of 8 and 16 bit MM using 4:2 compressor based MM is portrayed in Fig.15 and Fig.16.

| Device Utilization Summary | | |
|---|---|---|
| **Slice Logic Utilization** | **Used** | **Available** |
| Number of Slice Registers | 0 | 126,800 |
| Number of Slice LUTs | 55 | 63,400 |
| Number used as logic | 55 | 63,400 |
| Number using O6 output only | 45 | |
| Number using O5 output only | 0 | |
| Number using O5 and O6 | 10 | |
| Number used as ROM | 0 | |
| Number used as Memory | 0 | 19,000 |
| Number used exclusively as route-thrus | 0 | |
| Number of occupied Slices | 29 | 15,850 |

**Fig.15. Area utilization of 8 bit MM using 4:2 compressor based AM**

| Device Utilization Summary | | |
|---|---|---|
| **Slice Logic Utilization** | **Used** | **Available** |
| Number of Slice Registers | 3 | 126,800 |
| Number used as Flip Flops | 0 | |
| Number used as Latches | 0 | |
| Number used as Latch-thrus | 0 | |
| Number used as AND/OR logics | 3 | |
| Number of Slice LUTs | 244 | 63,400 |
| Number used as logic | 244 | 63,400 |
| Number using O6 output only | 196 | |
| Number using O5 output only | 0 | |
| Number using O5 and O6 | 48 | |
| Number used as ROM | 0 | |
| Number used as Memory | 0 | 19,000 |
| Number used exclusively as route-thrus | 0 | |
| Number of occupied Slices | 115 | 15,850 |

**Fig.16. Area utilization of 16 bit MM using 4:2 compressor based AM**

From Fig.15, it is observed that for 8 bit MM, it contains 55 LUTs with 29 slices occupied. Similarly for 16 bit MM, it includes 244 LUTs with 115 slices occupied as shown in Fig.16. It is inferred that the area is reduced by 39.5% for 8 bit and 43.9% for 16 bit when compared to 3:2 compressor based MM.

*iii. Area Analysis of 5:2 compressors based MM*

Here, the area utilization of 8 and 16 bit MM using 5:2 compressor based MM is depicted in Fig.17 and Fig.18.

| Device Utilization Summary | | |
|---|---|---|
| **Slice Logic Utilization** | **Used** | **Available** |
| Number of Slice Registers | 0 | 126,800 |
| Number of Slice LUTs | 48 | 63,400 |
| Number used as logic | 48 | 63,400 |
| Number using O6 output only | 34 | |
| Number using O5 output only | 0 | |
| Number using O5 and O6 | 14 | |
| Number used as ROM | 0 | |
| Number used as Memory | 0 | 19,000 |
| Number used exclusively as route-thrus | 0 | |
| Number of occupied Slices | 24 | 15,850 |

**Fig.17.Area utilization of 8 bit MM using 5:2 compressor based AM**

| Device Utilization Summary | | |
|---|---|---|
| **Slice Logic Utilization** | **Used** | **Available** |
| Number of Slice Registers | 1 | 126,800 |
| Number used as Flip Flops | 0 | |
| Number used as Latches | 0 | |
| Number used as Latch-thrus | 0 | |
| Number used as AND/OR logics | 1 | |
| Number of Slice LUTs | 213 | 63,400 |
| Number used as logic | 213 | 63,400 |
| Number using O6 output only | 176 | |
| Number using O5 output only | 0 | |
| Number using O5 and O6 | 37 | |
| Number used as ROM | 0 | |
| Number used as Memory | 0 | 19,000 |
| Number used exclusively as route-thrus | 0 | |
| Number of occupied Slices | 104 | 15,850 |

**Fig.18.Area utilization of 16 bit MM using 5:2 compressor based AM**

From Fig.17, it is observed that for 8 bit MM, it consists of 48 LUTs with 24 slices occupied. Similarly for 16 bit MM, it includes 213 LUTs with 104 slices occupied as shown in Fig.18. It is noted that the area is reduced by 47.2% for 8 bit and for 43.2% 16 bit when compared to 3:2 compressor based MM. Likewise, it is reduced by 12.7% for 8 bit and 12.7% for 16 bit compared to 4:2 compressor.

*C. Delay Analysis*

The delay obtained through 8 and 16 bit MM using 3:2, 4:2 and 5:2 compressors based AM is discussed below.

*i. Delay of 3:2 compressor based MM*

Here, the delay of 8 and 16 bit MM using 3:2 compressor based MM is shown in Fig.19 and Fig. 20.

```
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 6623329 / 16
-------------------------------------------------------------
Offset:             12.298ns (Levels of Logic = 18)
  Source:           y<4> (PAD)
  Destination:      z2_7 (FF)
  Destination Clock: clk rising
```

**Fig.19. Delay of 8 bit MM using 3:2 compressor based AM**

```
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 457715052372071 / 32
-------------------------------------------------------------
Offset:             28.706ns (Levels of Logic = 64)
  Source:           y<4> (PAD)
  Destination:      z2_31 (FF)
  Destination Clock: clk rising
```

Fig.20. Delay of 16 bit MM using 3:2 compressor based AM

From Fig.19 and Fig.20 ,it is observed that the delay of 8 and 16 bit MM using 3:2 compressor based AM is 12.298 ns and 28.706 ns respectively.

*ii. Delay of 4:2 compressor based MM*

Here, the delay of 8 and 16 bit MM using 4:2 compressors based MM is illstrated in Fig.21 and Fig.22.

```
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 150551 / 16
-------------------------------------------------------------
Offset:             6.890ns (Levels of Logic = 11)
  Source:           x<0> (PAD)
  Destination:      z2_5 (FF)
  Destination Clock: clk rising
```

**Fig.21.Delay of 8 bit MM using 4:2 compressors based AM**

```
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 503164321 / 32
-------------------------------------------------------------
Offset:             16.622ns (Levels of Logic = 44)
  Source:           y<3> (PAD)
  Destination:      z2_31 (FF)
  Destination Clock: clk rising
```

**Fig.22. Delay of 16 bit MM using 4:2 compressors based AM**

From Fig.21 and Fig.22 it is observed that the delay of 8 and 16 bit MM using 4:2 compressor based AM is 6.89 ns and 16.622 ns. It is also inferred that the delay is reduced by 34.9% for 8 bit and 42% for 16 bit when compared to 3:2 compressor based MM.

*iii. Delay of 5:2 compressor based MM*

Here, the delay of 8 and 16 bit MM using 3:2 compressors based MM is illustrated in Fig.23 and Fig.24.

```
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 48454 / 16
-------------------------------------------------------------
Offset:             5.892ns (Levels of Logic = 10)
  Source:           x<0> (PAD)
  Destination:      z2_5 (FF)
  Destination Clock: clk rising
```

**Fig.23. Delay of 8 bit MM using 5:2 compressor based AM**

```
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 17876339 / 32
-------------------------------------------------------------
Offset:             14.595ns (Levels of Logic = 42)
  Source:           y<3> (PAD)
  Destination:      z2_31 (FF)
  Destination Clock: clk rising
```

**Fig.24. Delay of 16 bit MM using 5:2 compressor based AM**

From Fig.23 and Fig.24, it is observed that the delay of 8 and 16 bit MM using 5:2 compressor based AM is 5.892 ns and 14.595 ns. It is verified through the delay analysis that the delay is reduced by 52% for 8 bit and for 49.1% 16 bit when compared to 3:2 compressor based MM. Likewise, it got reduced by 14.4% for 8 bit and 12.1% for 16 bit compared to 4:2 compressor.

## VII. PERFORMANCE COMPARISON

In this section, the performance of MM using 3:2, 4:2 and 5:2 compressors based AM are compared in terms of area and delay with the existing MM [5]. For better understanding, the theoretical analysis of compressor based MM is compared in theoretically and the results are tabulated below in Table-I.

**Table-I: Theoretical Analysis of MM Using Compressors based AM**

| Parameters | 8 bit MM using compressors based AM | | | 16 bit MM using compressors based AM | | |
|---|---|---|---|---|---|---|
| | 3:2 | 4:2 | 5:2 | 3:2 | 4:2 | 5:2 |
| No. of Stages | 7 | 3 | 2 | 15 | 5 | 3 |
| No. of FA | 48 | 45 | 38 | 224 | 155 | 144 |
| No. of HA | 8 | 6 | 4 | 16 | 10 | 6 |

From the Table-I, it is inferred that the MM using 5:2 compressors based AM outperforms the MM using 3:2 and 4:2 compressors based AM theoretically because the number of stages and adders required is also low in proposed MM.

**Table-II: Comparative results of MM using Compressors based AM**

| Parameters | Slice LUTs | | Slices occupied | | Delay (ns) | |
|---|---|---|---|---|---|---|
| | 8 bit | 16 bit | 8 bit | 16 bit | 8 bit | 16 bit |
| [5] | 138 | 590 | 73 | 304 | 38.801 | 78.832 |
| 3:2 | 91 | 375 | 34 | 179 | 12.298 | 28.706 |
| 4:2 | 55 | 244 | 29 | 115 | 6.89 | 16.622 |
| 5:2 | 48 | 213 | 24 | 104 | 5.892 | 14.595 |

From the Table-II, it is shown that the 8 and 16 bit MM using 5:2 compressors based AM gives better results in terms of area and delay compared to MM using 3:2 and 4:2 compressors based AM and also the existing method [5]. The comparison of existing MM [5] with proposed compressor based MM using array multiplier in terms of percentage in area and delay reduction is tabulated below.

**Table-III: Reduction Percentage Analysis**

| Parameters | Area Reduction (%) | | Delay Reduction (%) | |
|---|---|---|---|---|
| | 8 bit | 16 bit | 8 bit | 16 bit |
| 3:2 | 34 | 36.4 | 68.3 | 63.5 |
| 4:2 | 60 | 58.6 | 82.2 | 78.9 |
| 5:2 | 65.21 | 63.8 | 84.8 | 81.4 |

Here, it is inferred through Table III that performance comparison is done between existing work [5] and proposed compressor based MM in terms of percentage. The reduction in percentage of delay and area obtained through the proposed method proves that the proposed 5:2 compressors based MM gives better performance than 3:2 and 4:2 compressors based MM in all attributes when compared to existing MM [5] as shown in Table-III.

## VIII.CONCLUSION

In this paper, Montgomery modular multiplication using compressors based array multiplier is developed and compared. Montgomery Multiplier using 3:2, 4:2 and 5:2 compressors based array multiplier is developed and the performance are compared in terms of area and delay in the device Artix-7. From the results, it is observed that MM using 5:2 compressors based AM achieves better performance in terms of area and propagation delay compared to that of MM using 3:2 and 4:2 compressors. The proposed AM using 5:2 compressors can be used for higher order multiplications to improve the performance of the MM. And also higher order (i.e.) n:2 compressors based AM can also be implemented. Further, the improvised MM can be employed in ECC and RSA cryptographic algorithms to promote their running speed.

## REFERENCES

1. S. Khan, K. Javeed, Y.A. Shah, "High-Speed FPGA Implementation of Full-Word Montgomery Multiplier for ECC Applications", Microprocessors and Microsystems, Vol.62, pp 91-101, October 2018.
2. T. U. S. Krishna, K. S. Riyas, Y. Premson, and R. Sakthivel, "15–4 Approximate Compressor Based Multiplier for Image Processing", Proceedings of 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, pp. 671-675, May 2018.
3. S. Venkatachalam and S.B.Ko, "Design of Power and Area Efficient Approximate Multipliers", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.25, pp.1782-1786, January 2017.
4. T. Adiono, H. Ega, H. Kasan,S. Fuada, S. Harimurti,"Full custom design of adaptable Montgomery Modular Multiplier for asymmetric RSA cryptosystem", Proceedings of the International Symposium on Intelligent Signal Processing and Communication System (ISPACS), Xiamen, China, pp.910-914, November 2017.
5. C.Anoop, Anu Chalil, "Performance analysis of Montgomery Multiplier" Proceedings of the 2nd International Conference on Communication and Electronics Systems (ICCES)", Coimbatore, pp.26-29, October 2017.
6. Y. Mo, S. Li, "Design of an 8192-bit RNS Montgomery multiplier," Proceedings of the International Conference on Electron Devices and Solid-State Circuits (EDSSC)", Taiwan, pp.1-2, October 2017.
7. S.R. Kuang, K.Y. Wu, R.Y. Lu, "Low-cost high-performance VLSI architecture for Montgomery Modular Multiplication", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.24, No. 2 ,pp. 434–443, February 2016.
8. K. Javeed, D. Irwin, X. Wang, "Design and performance comparison of modular multipliers implemented on FPGA platform", Proceedings of Springer International Conference on Cloud Computing and Security (ICCS),China, pp. 251–260, November 2016.
9. N. Thampi, M. E. Jose, "Montgomery Multiplier for Faster Cryptosystems", Proceedings of Elsevier 1st Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST),Kerala Vol. 25, pp. 392–398, September 2016.
10. A.Thomas, E. M. Manuel, "Embedment of Montgomery Algorithm on Elliptic Curve Cryptography over RSA Public Key Cryptography", Proceedings of Elsevier International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST), Kerala, Vol. 24, pp. 911–917, July 2016.
11. M. Morales-Sandoval, A. Diaz-Perez, "Scalable gf (p) Montgomery multiplier based on a digit–digit computation approach", IET Computers & Digital Techniques, Vol. 10, No. 3, pp.102–109, April 2016.
12. G. S. Lakshmi, D. Fatima, and B. K. Madhavi, "Compressor based 8×8 BIT vedic multiplier using reversible logic", Proceedings of 3rd International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, pp.174-178, September 2016.

13. N. Mulla, A. Kasetwar, "FPGA implementation of an efficient Montgomery Multiplier for adaptive filtering application", Proceedings of IEEE International Conference on Power, Automation and Communication (INPAC), Amravati pp.66-70, October 2014.
14. X. Yan, G. Wu, D. Wu, F. Zheng, X. Xie," An implementation of Montgomery modular multiplication on FPGAs", Proceedings of IEEE International Conference on Information Science and Cloud Computing (ISCC), China, pp. 32–38, December 2013.
15. T. Wu, "Improving radix-4 feedforward scalable Montgomery modular multiplier by precomputation and double booth-encodings," Proceedings of IEEE 23rd International Conference on Computer Science and Network Technology (ICCSNT)", China, pp. 596-600, October 2013.
16. S. Kakde, G. Somulu, P. Zode, "Performance analysis of Montgomery multiplier for public key cryptosystem," Proceedings of 4th IEEE International Conference on Computing, Communications and Networking Technologies (ICCCNT)", Tiruchengode, pp.1-5, July 2013.
17. A. Pishvaie, G. Jaberipur, and A. Jahanian, "Redesigned CMOS (4; 2) compressor for fast binary multipliers," Canadian Journal of Electrical and Computer Engineering,Vol. 36, no. 3, pp. 111-115,September 2013.
18. A. Pishvaie, G. Jaberipur, and A. Jahanian, "Improved CMOS (4;2) compressor designs for parallel multipliers", Computers & Electrical Engineering, Vol. 38, no. 6, pp. 1703–1716, November 2012.
19. G.C Chow, K. Eguro, W. Luk, P. Leong, "A Karatsuba-based Montgomery multiplier", Proceedings of IEEE International Conference on Field Programmable Logic and Applications (FPL), Italy, pp.434-437, September 2010.
20. Y. Gong,S. Li, "High-Throughput FPGA Implementation of 256bit Montgomery Modular Multiplier ", Proceedings of IEEE Second International Workshop on Education Technology and Computer Science, China, pp. 173-176, March 2010.
21. S. Gao, D. Al-Khalili, and N. Chabini, "Implementation of large size multipliers using ternary adders and higher order compressors", Proceedings of International Conference on Microelectronics - ICM, Marrakech, pp.118-121, December 2009.
22. S. Veeramachaneni, K. Krishna, L. Avinash, S. Puppala, and M. Srinivas, "Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors", Proceedings of IEEE 20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID07), Bangalore,pp. 324-329,January 2007.

## AUTHORS PROFILE

**G.Revathi** has recently completed M.Tech degree in the Department of Electronics Engineering, Pondicherry University, Puducherry, India in the year 2019. She is extensively working in the area of Digital Electronics and Digital VLSI. Email: pathyrevathi140@gmail.com

**K V Gowreesrinivas** is a Research Scholar of Department of Electronics Engineering of Pondicherry University, Puducherry, India. He is the Life time Member of *IETE*. He is extensively working in the area of Microelectronics and Digital VLSI. Email: srinu43306@gmail.com

P. **Samundiswary** is working as Assistant Professor of Department of Electronics Engineering in Pondicherry University, Pondicherry, India. She has received her B.Tech and M.Tech degrees in the field of Electronics and Communication Engineering from Pondicherry Engineering College affiliated to Pondicherry University. She obtained her Ph. D degree in the field of Electronics and Communication Engineering from Pondicherry Engineering College affiliated to Pondicherry University, Pondicherry, India in 2011. She has nearly 20 years of experience in teaching profession. She is a author of more than 90 papers in national and international conference proceedings and journals. She has been one of the authors of the book published by LAMBERT Academic Publishing. Also, she has co-authored around five book chapters published by INTECH and Springer Publishers. Her area of interest includes Wireless Communication and Networks, Wireless Security and Computer Networks. She will be available at samundiswary_pdy@yahoo.com.