International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

An Efficient Vlsi Architecture For Montgomery Modular Multiplier

Mrs . N . Bhargavi & Ms. Kapuluru Leelavathi
1 PG Student, Dept. Of VLSI and Embedded systems, SKR College Of Engineering & Technology, AP
2 Asst. Professor, Dept. Of VLSI and Embedded systems, SKR College Of Engineering &Technology,AP.

ABSTRACT:

Montgomery modular multiplication is used in cryptographic algorithms and digital signal processing application. The main objective is to reduce the delay and area of the Montgomery multipliers while maintaining low hardware complexity. To speed up, high-speed Montgomery modular multiplication algorithms and hardware architectures employ carry-save addition to avoid the carry propagation at each addition operation of the add-shift loop but it requires extra clock cycles and it increases hardware complexity. A Configurable CSA (CCSA) is proposed to for performing modular multiplication by using two serial half-adders and a mechanism that can detect and skip the unnecessary carry-save addition operations thereby maintaining the short critical path delay is developed by means of designing a skip detector. Simulation is carried out using Xilinx ISE Design Suite 13.2. The proposed Montgomery modular multiplier can achieve higher performance and high speed when compared conventional modular multiplier.

Keywords— Configurable Carry Save Adder; Skip detector;

1. INTRODUCTION:

Modular Multiplication is the central operation in many application areas including public key cryptography for encryption and decryption. The widely used method for modular multiplication is Montgomery modular multiplier. In which there will be a carry save adder . X′Y mod M is the operation to be performed. In which X and Y are the inputs. It is necessary to find the value of mod M, henceforth going for this algorithm. Comparing all previously occurring algorithms, this algorithm will produce the optimized output. There are two cases, semi carry save addition and full carry save addition. In this semi carry save addition, the given inputs are in binary and the inter outputs alone in carry save. Whereas in full carry addition, both inputs and inter outputs are in carry save.

On comparing, it can be seen that semi carry save is the most advantageous one because it has only one carry save and hence it has less area and high speed which is required for designing an VLSI based multipliers.

Thereby the circuit design will have a reconfigurable carry save adder, through which the expected throughput with high complexity can be obtained with the generation of 32 bits .

2. PREVIOUSLY PROPOSED ARCHITUCTURE

2.1. Montgomery Modular Multiplier

In propose a new SCS-based Montgomery MM algorithm to reduce the critical path delay of Montgomery multiplier. In addition, the drawback of more clock cycles for completing one multiplication is

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

also improved while maintaining the advantages of short critical path delay and low hardware complexity [2].

2.2. Critical Path Delay Reduction

The critical path delay of SCS-based multiplier can be reduced by combining the advantages of FCS-MM-2 and SCS-MM-2. That is pre compute $D = B + N$ and reuse the one-level CSA architecture to perform $B+N$ and the format conversion. Figure.1 shows the modified SCS-based Montgomery multiplication (MSCS-MM) algorithm and possible hardware architecture, respectively [3].

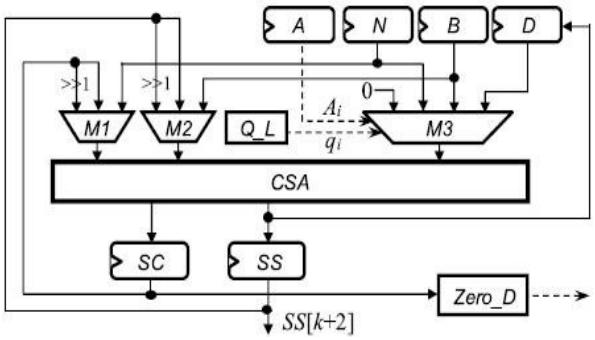


FIG.1. Diagram of Montgomery Modular Multiplier

Therefore, the critical path delay of the MSCS-multiplier can be reduced into TMUX4 + TFA. However, in addition to performing the three-input carry-save additions $k + 2$ times, many extra clock cycles are required to perform $B + N$ and the format conversion via the one-level CSA architecture because they must be performed once in every MM.

Furthermore, the extra clock cycles for performing $B+N$ and the format conversion through repeatedly executing the carry-save addition $(SS, SC) = SS+SC+0$ are dependent on the longest carrypropagation chain in $SS + SC$. If $SS = 111\ldots1112$ and $SC = 000\ldots0012$, the one-level CSA architecture needs k clock cycles to complete $SS + SC$. That is, $3k$ clock cycles in the worst case are required for completing one MM. Thus, it is critical to reduce the required clock cycles of the MSCS-MM multiplier [1].

2.3. Clock Cycle Number Reduction

To decrease the clock cycle number, a CCSA architecture which can perform one three- input carry-save addition or two serial two-input carry-save additions is proposed to substitute for the one-level CSA architecture [4].

Two cells of the one-level CSA architecture in Figure.2each cell is one conventional FA which can perform the three-input carry-save addition. Two cells of the proposed configurable FA (CFA) circuit. If α = 1, CFA is one FA and can perform one three-input carry-save addition (denoted as 1F_CSA).

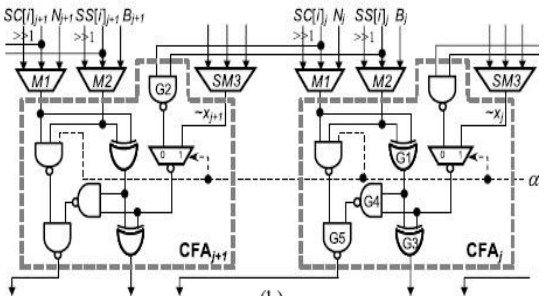


FIG.2. Carry Full Adder Circuit

The Zero_D circuit is used to detect whether SC is equal to zero, which can be accomplished using one NOR operation. The Q_L circuit decides the qi value.The carry propagation addition operations of B N and the format conversion are performed by

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

theone-level CSA architecture of the MSCS-MM multiplier through repeatedly executing the carry-save addition (*SS*, *SC*) = *SS* + *SC* + 0 until *SC* = 0.In addition, we also pre compute *Ai* and *qi* in iteration *i*−1 (this will be explained more clearly in Section III-C) so that they can be used to immediately select the desired input operand from 0, *N*, *B*, and *D* through the multiplexer *M3* in iteration *I [5]* .

Otherwise, it is two half-adders (HAs) and can perform two serial two-input carry-save additions (denoted as 2H_CSA). In this case, G1 of CF Aj and G2 of CFAj+1 will act as HA1 j and G3, G4, and G5 of CF Aj will behave as HA2j.

Moreover, we modify the 4-to-1 multiplexer M3 into a simplified multiplier SM3 because one of its inputs is zero, where the INVERT operation. Note that M3 has been replaced by SM3 in the proposed one-level CCSA architecture.

3.SCS-MM2 Algorithm

The modified SCS-MM2 algorithm is shown in fig.1. Initially we make the carry and sum values as the sum of multiplier and the modulus this is pre-computation step. The steps from 3 to 4 iterates for K times. Here K represents the number of bits and i represents i^{th} bit. In fig.1 suffix 0 represents the least significant bit.

```
Algorithm MM:
Modified SCS-MM2 algorithm

Input : A, B, N ( modulus )
Output : S[ k ]
1. ( SS [0], SC [ 0 ] ) = (B + N + 0)
2. For ( i = 0 to k - 1 )
3. {  q [ i ] = ( SS [ i ] ) + A [ i ] * B ) mod2;
      If ( A [ i ] = 0  and  q [ i ]0 =  0 )  X = 0;
      If ( A [ i ] = 0  and  q [ i ]0 =  1 )  X = N;
      If ( A [ i ] = 1  and  q [ i ]0 =  0 )  X = B;
      If ( A [ i ] = 1  and  q [ i ]0 =  1 )  X = B + N;
4.    SS[i+1 ] , SC[ i + 1]= ( SC [ i ] + SS [ i ] + X ) / 2;
   }
5. If ( SS[ k ] >= N)  then
6.          S[ k ] =SS[ k ] – N;
7. else     return S[ k ];
```

Fig3. Modified SCS-MM2 algorithm

Adders are of many types. Out of those carry save adder is efficient because it is having less propagation delay. Carry Save adder for n-bit means it is having n-parallel adders, which produce n-bit sums and n-bit carry's. The inputs for carry save adder are SS,SC and mux output. Mux output depends up on "aa" and "qa" of a single bit. Here we considered "aa" as A[i] * B and gate Least Significant Bit. "qa" represents the sum of SS and "aa".

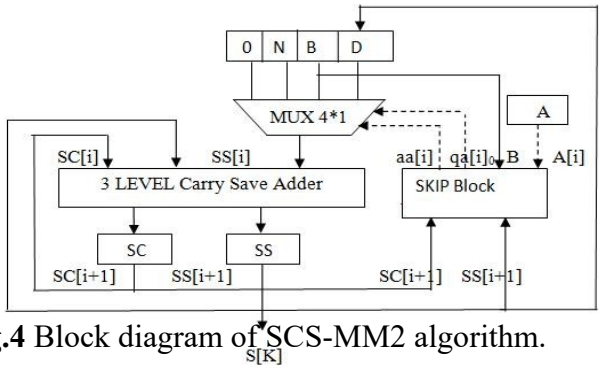
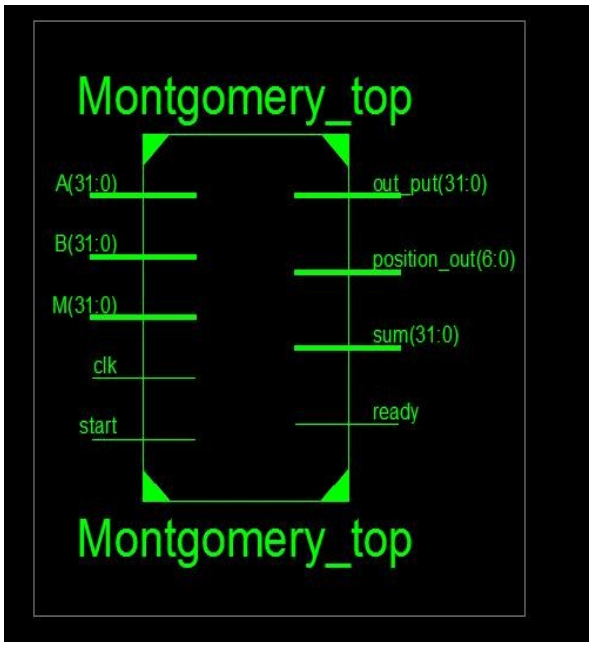


Fig.4 Block diagram of SCS-MM2 algorithm.

In fig2 depending upon "aa" and "qa" values the third input for the CSA varies. This loop iterates for n times. The final stage sum is considered as the final output. The CSA block internally consists of full adders.

4.Simulation results:



International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

Fig_5 Block diagram:

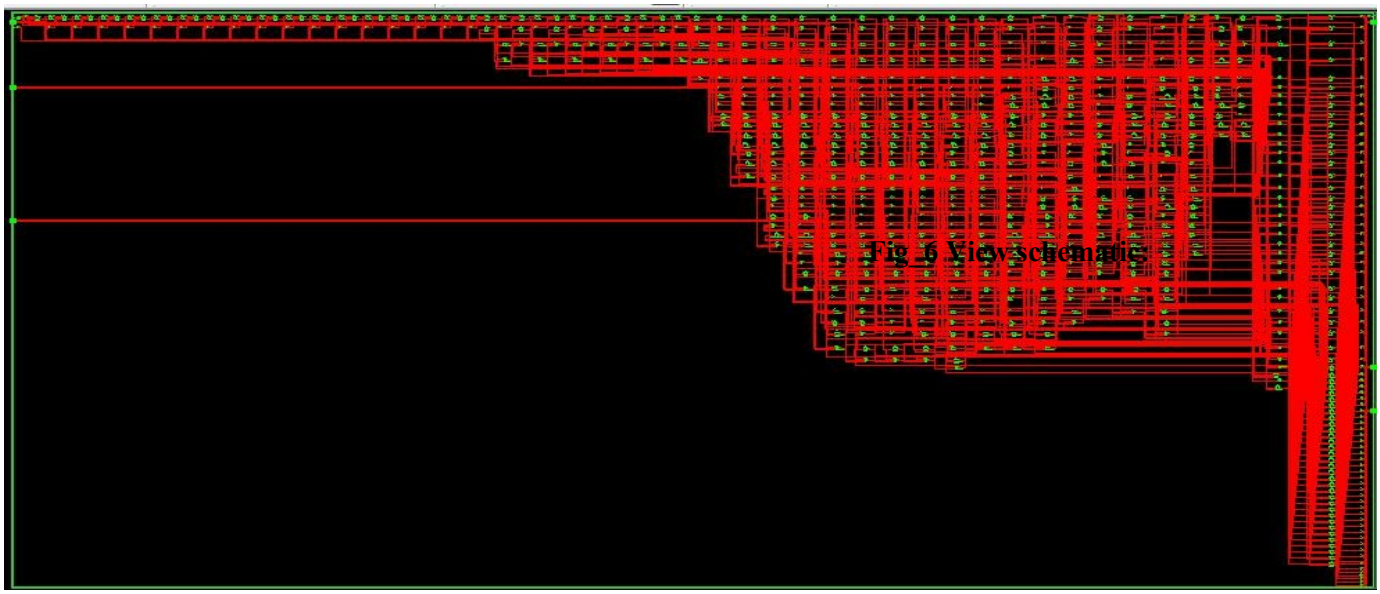


Fig_6 View schematic:

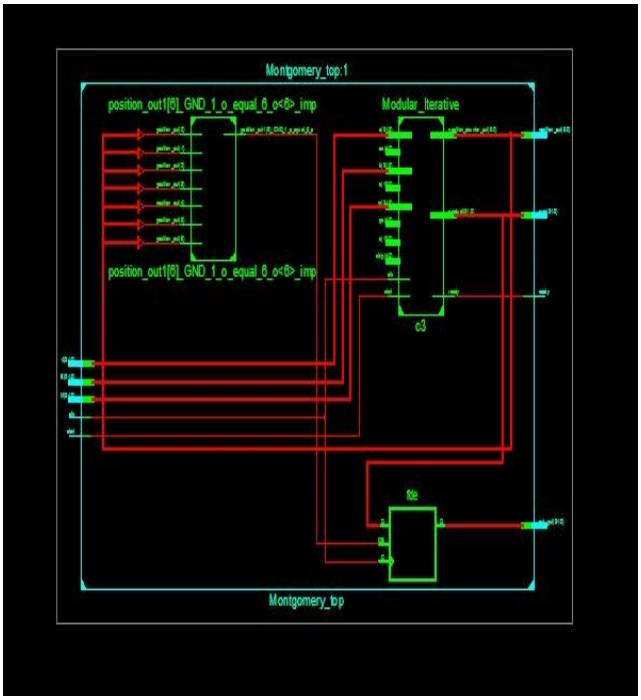


Fig_6 RTL Schematic:

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

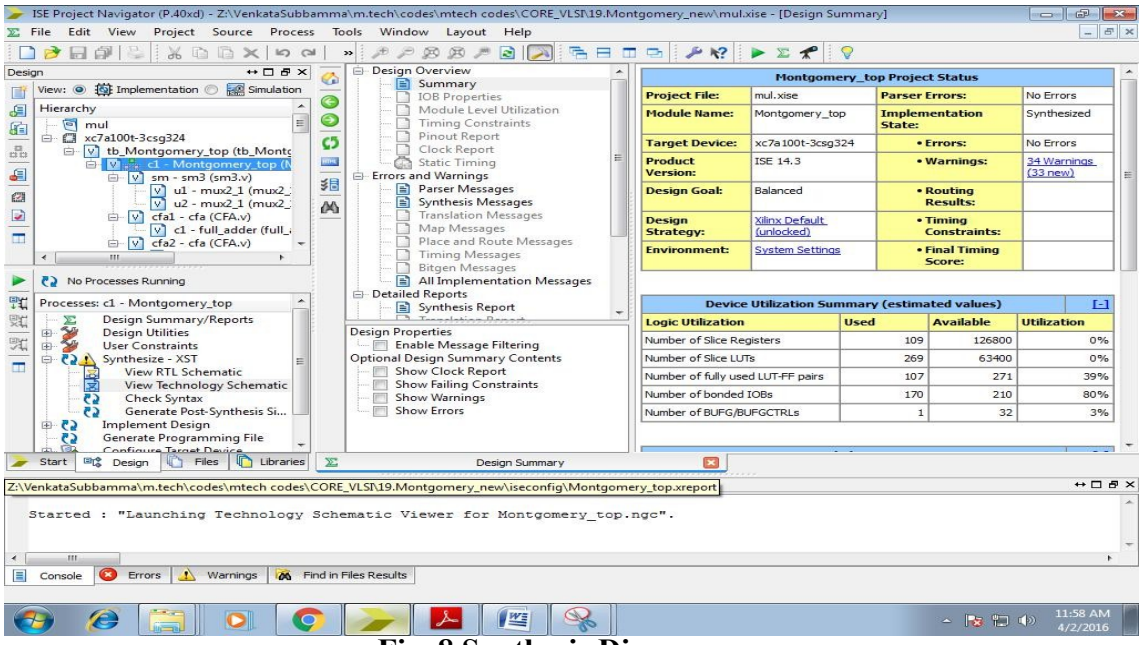


Fig_8 Synthesis Diagram

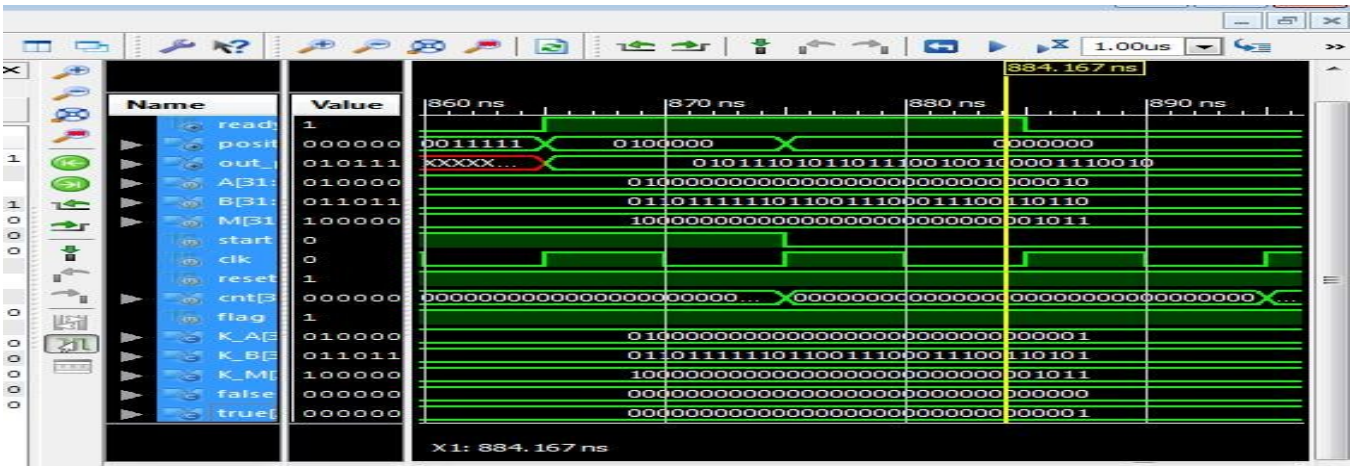


Fig_9 Result

5.CONCLUSION

FCS-based multipliers maintain the input and output operands of the Montgomery MM in the carry-save format to escape from the format conversion, leading to fewer clock cycles but larger area than SCS-based multiplier. To enhance the performance of Montgomery MM while maintaining the low hardware complexity, this paper has modified the SCS-based Montgomery multiplication algorithm and pro-posed a low-cost and high-performance Montgomery modular multiplier. The proposed multiplier used one-level CCSA architecture and skipped the unnecessary carry-save addi-tion operations to largely reduce the critical

path delay and required clock cycles for completing one MM operation. Experimental results showed that the proposed approaches

are indeed capable of enhancing the performance of radix-2 CSA-based Montgomery multiplier while maintaining low hardware complexity.

REFERENCES

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[2] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances inCryptology*. Berlin, Germany: Springer-Verlag, 1986, pp. 417–426.

[3] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[4] P. L. Montgomery, "Modular multiplication without trial division," *Math.Comput.*, vol. 44, no. 170, pp. 519–521, Apr. 1985.

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

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

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

[8] Y.-Y. Zhang, Z. Li, L. Yang, and S.-W. Zhang, "An efficient CSA architecture for Montgomery modular multiplication," *MicroprocessorsMicrosyst.*, vol. 31, no. 7, pp. 456–459, Nov. 2007.

[9] C. McIvor, M. McLoone, and J. V. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques," *IEE Proc.-Comput. Digit. Techn.*, vol. 151, no. 6, pp. 402–408, Nov. 2004.

[10] S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, "Energy-efficient high-throughput Montgomery modular multipliers for RSA cryptosys-tems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 11,
pp. 1999–2009, Nov. 2013.

Author's Profile:

Mrs.N.BHARGAVIreceived B.Tech in Electronics and Communication Engineering from Narayana Engineering College, Nellore affiliated to the Jawaharlal Nehru technological university Anatapur in 2011, and pursing M. Tech in VLSI and Embedded systems from SKR College of Engineering

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

affiliated to the Jawaharlal Nehru technological university Anantapur in 2017, respectively.

Ms. KAPULURU LEELAVATHI as Asst Professor Department of ECE.Qualification: M.Tech SKR College of Engineering & Technology
Email ID: leelavathi256@gmail.com