ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

Implementation of Optimized Montgomery modular Multiplier on FPGA

Dr.Deepa Jose[1], Nathimugil.J[2], Abida Begum[3]
Associate Professor, Centre for Excellence in VLSI, Department of Electronics and Communication Engineering, KCG College of Technology, Chennai, Tamil Nadu, India[2]
Email-deepa.ece@kcgcollege.com
PG Scholar, Centre for Excellence in VLSI, Department of Electronics and Communication Engineering, KCG College of Technology, Chennai, Tamil Nadu, India[1]
Email-nathimugil03@gmail.com
Assistant Professor, Centre for Excellence in VLSI, Department of Electronics and Communication Engineering, KCG College of Technology, Chennai, Tamil Nadu, India[2]

***Abstract*— In data transmission applications, the widely used public-key cryptosystem is RIVEST-SHAMIR-ADLEMAN which includes encryption and decryption process. In order to attain higher speeds, many algorithms and architectures that use carry-save addition to avoid the carry propagation at each addition operation of the add-shift loop is employed. Due to complexity in terms of time and resources needed, the RSA system becomes slow and difficult to implement and to overcome this, a speed-efficient algorithm with a configurable Carry save adder with its architecture is proposed. It consists of ALU which includes carry save adder and multiplier leading to higher throughput. Experimental results show that the proposed approach can exhibit improved performance for 63-bit Montgomery multiplier.**

***Index Terms*— carry-save addition, cryptography, Montgomery modular multiplier, speed.**

I. 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 [1-5].

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. Comparingall 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 expectedthroughput with high complexity can be obtained with the generation of 32 bits [6-7]..

II. EXISTING MONTGOMERY MULTIPLIER

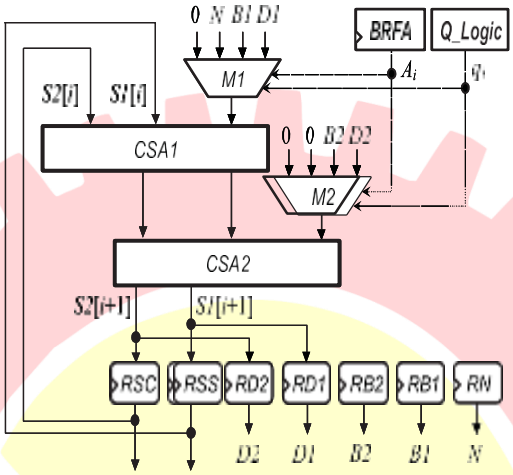
Consider the modulus N to be a k-bit odd number and an extra factor R is to be defined as 2k mod N, where $2k-1 \leq N < 2k$. Given two integers a and b, where a, b <N, the N–residue of a and b with respect to R can be defined as,

$A = a \times R \pmod{N}$ (1)

$B = b \times R \pmod{N}$ (2)

Based on equation (1), the Montgomery modular product Z of A and B can be obtained as

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016



$$Z = A \times B \times R^{-1} \pmod{N} \qquad (3)$$

Fig.1. MM42 multiplier

In this MM42 multiplier, there will be two carry save adders. One CSA is used to perform the primary operation whereas the second CSA is used to perform the later operations. Hence there will be more complexity because of its greater time consumption and therefore less speed[8-10].

The operation behind this is that the inputs and the modulus are allowed inside the multiplexer. That partial product will tentatively allow inside carry save adder1. This product is then allowed inside the carry save adder2. Then those partial outputs are separated, and then the required output can be obtained [11].

The recent found says that this type of architecture causes complexity as well as Multiplicity. As long as the carry save adder gets increased, the number of iterations also increases.

The operation behind this feedback loop causes time -complexity. Though much iterations are found in this architecture hence going for the proposed system.

III. PROPOSED MONTGOMERY MULTIPLIER

In this proposed system, carry save addition with semi-carry approach is described. In which all the multiplicands are not recycled, that is whatever the multiplicand is needed to be multiplied at that time alone is used for determining the output.

The carry save approach has higher benefits since it is the basic key for operating a Montgomery modular multiplier. In such a way, using this semi carry save type only one carry level adder is implemented which may be two serial half adders or a full adder can be used based on the requirement.

It thereby reduces the number of clock cycles and hence less delay. So the output will be optimized and it can be implemented using Verilog coding.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

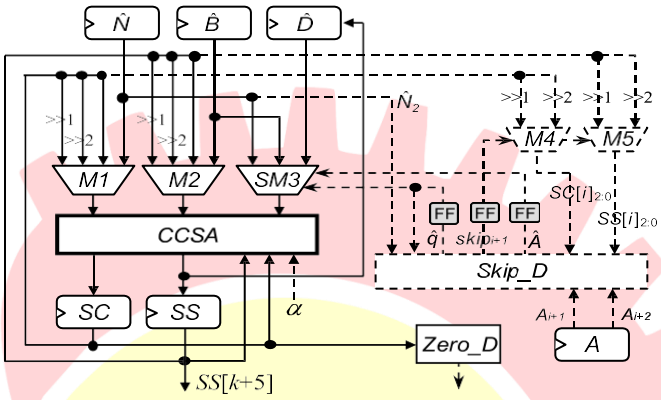


Fig. 2 Block diagram of proposed multiplier

The above architecture is the semi-carry save based Montgomery multiplier. In which the loop is reduced on comparing to the existing one. It consists of two multiplexers, one multiplier, one configurable carry save adder, flip-flops, skip detector and zero detector.

A. Architecture of proposed Multiplier

Fig. 2 illustrates the block diagram of proposed semi carry save multiplier. It is first used to pre-compute the four-to-two carry save additions. Then the required multiplication can be performed. The modulus N and inputs will be allowed inside the two multiplexers. This partial product is then allowed inside the multiplier.

Those partial outputs then enter into configurable carry save adder, where the carry save addition operation is performed. They are stored in the flip flops temporarily. When another partial output is executed, then that will be stored in the flip flop.

The Skip detector will skip the previous multiplication which is not required in the operation so as to reduce the number of clock cycles. The partial product from SM3 is allowed to the multiplexers M4 and M5. Later on it allows inside the flip flops for temporary storage, then to the skip detector.

The output can be obtained from semi carry. This process is repeated until the output is obtained. The zero detectors can also be used to detect zero in many situations, which is most required. The complexity is very less compared to the previous one.

B. Critical Path Delay Reduction

In order to reduce the critical path delay, the operations in semi carry save and full carry save is performed jointly. The carry save format conversion as well as the binary format should be taken place. Then pre computation must be done in order to reuse the multiplicand values.

Another method is using zero detectors. SC will produce the output only when the zero is detected. Then the pre computation can be done i-1 iteration.

C. Clock cycle number reduction

In order to decrease the clock cycle number, a configurable carry save architecture to perform one three-input carry-save addition or two serial two-input carry-save addition is used.

Furthermore the number of iteration can also be reduced to reduce number of clock cycles.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

Then a signal skip is used. In order to verify whether i+1 is required or not to be happen in the upcoming events. This can be found in the previous i iteration itself.

By again using the same condition, signal skip will use i+1, so that it increases by a factor 2. Hence it directly goes to i+2. So that clock cycle gets reduced.

D. Quotient pre computation

A_{i+1}, A_{i+2}should be known already in order that the unwanted steps in the (i +1) iteration can be reduced by determining i iteration. So as to pre compute the quotients.

Another method is using skip detector so that it will pre computes the values. And also since the shortest path in this multiplier is lengthened, it has to be minimized.

As modulus N is an odd number, it can be used directly for the multiplication. So that time is consumed highly.

IV. RESULTS AND DISCUSSION

By adopting semi carry save addition using SPARTAN 3a, the Delay, Area, and Clock Cycle Number can be realized and the waveform is also shown.

A. Analysis of Delay, Area, and Clock Cycle Number

For a semi carry save strategy, the maximum delay obtained is $2.71 \times T_{FA}$.. From the below table, it can be seen that the maximum delay of SCS multiplier for 1024 bits is 4.01 and for 2048 bits is 4.40. Hence Thereby on comparing to all other Montgomery multipliers the delay is reduced.

TABLEI
COMPARISON OF SCS MULTIPLIERWITH1024 AND2048 BIT KEYSIZES

Key size	Multiplier	Delay	Area	No. of clock Cycles
1024	SCS	4.01	498378	876
2048		4.40	950180	1732

The area complexity will be obtained as$9.88k \times A_{FA}$.It will be 498378 for 1024 bits and 950180 for 2048 bits. This minimized area usageis due to the configurable carry save adder which will pre propagate and then reuse the values.

On analyzing Clock cycle number, semi carry save offer cycle number reductions of 14.5% over other multipliers. It offers least clock cycles because extra clock cycles for format conversion is not required.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

B. Implementation Results

The design has been implemented using Xilinx Verilog coding. For further verification, the design can be done using Cadence. It can be clearly understand by the waveform shown below. It can be proven that it has reduced area complexity and speed complexity on comparing to all other multipliers.

The method has been implemented using a configurable carry save adder so as to prove the maximum delay to be less comparing all.

The delay and area can be minimized as much as possible as comparing to all other previous existing architectures.

The circuit diagram is obtained by implementing Verilog coding. It consists of clock, reset, and address, write data and read data. The output can be obtained at the read data. For each modulus, the result will be unique. The address can also be specified so as to identify the result separately.

The exponentiation can also be applied. Clock and the reset is tentatively applied to one. Modular multiplier produces the output by using validation inputs.

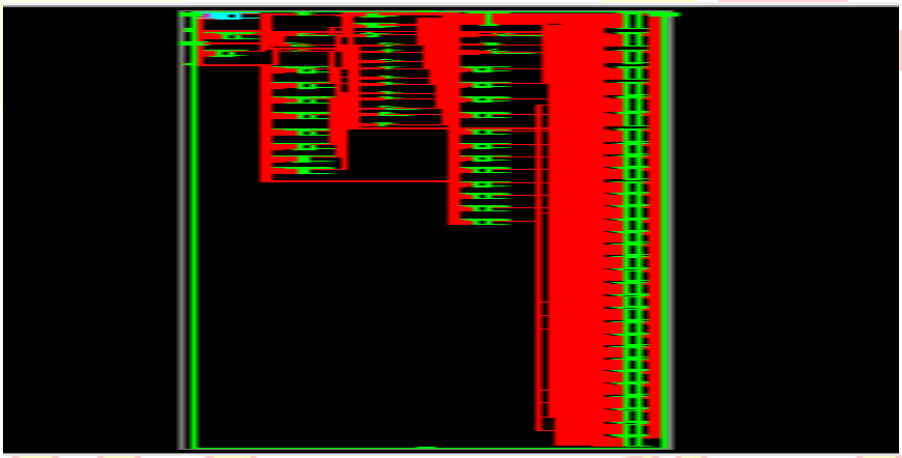


Fig 3 RTL schematic diagram of modular multiplier

The above diagram is the RTL schematic of the modular multiplier.it consists of many configurable carry save adders, flip flops, multiplexers and multipliers.

There will be three input AND gates. Modulus exponent length will also be there. So that modulus value can be applied. Since many interconnection wires are there, it is quite difficult to determine whether it is three inputs or four inputs.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

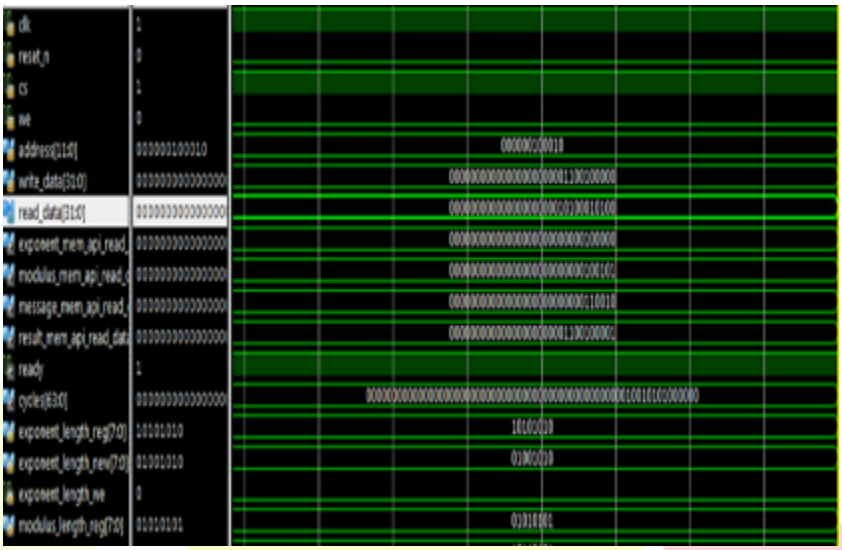


Fig 4 Output Waveform

The above waveform is the semi carry save output waveform which is obtained using XilinxVerilog software. The result shows that it has high speed and less area. Thereby it is proven that optimized Montgomery modular multiplier has been designed.

It can be seen that there is a clock, reset, write data and read data. Some other additional parameters are also found such as exponent, modulus, result, ready, cycles and modulus length also has been identified.

Fourth waveform is that one which the input can be given, fifth waveform consists of modulus and in the later one, the output can be noted. It can be obtained in binary, hexadecimal and decimal.

This exponent is the additional parameter which is highly required for various purposes. Since the exponent to the power in decimal in anything is equal to one, this method sometimes becomes unsuitable.

Whenever a high speed and low area is required, this architecture can be applied. Furthermore, this waveform enables that there is a little iteration in each step.

So that determining the requirements tends to produce as accurate as possible. By modifying the carry save adder to a reconfigurable one, this can be achieved.

V. CONCLUSION

In this paper, semi carry save adder is described. In order to obtain less area and high speed, semi carry save based Multiplier has been implemented. Though full carry save has high speed but covers higher area, this method is adopted. In which the less area also has been designed using configurable carry save adder. Furthermore, the experimental results show that it has reduced number of clock cycle and reduced delay.

VI. ACKNOWLEDGMENT

I would like to express my sincere thanks to Late. Dr. AMOS H. JEEVA OLI, Former Head of Department,ECE, KCG college of Technology,Chennai for being a project supervisor and for giving me valuable directions and Wonderful technical guidance throughout this paper.

REFERENCES

[1] Shiann-Rong Kuang, Member, Kun-Yi Wu and Ren-Yao Lu, June 2015 "Low-Cost High-Performance VLSI Architecture for Montgomery Modular Multiplication".,vol.22, no. 8,pp. 640–650

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

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

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

[5] J. Han, S. Wang, W. Huang, Z. Yu, and X. Zeng, , Dec.2013"Parallelization of radix-2 Montgomery multiplication on multicore platform," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 12, pp. 2325–2330.

[6]F. Gang, Jun. 2006, "Design of modular multiplier based on improved Montgomery algorithm and systolic array," in Proc. 1st Int. Multi-Symp. Comput.Comput.Sci., vol. 2.pp.356–359.

[7] G. Sassaw, C. J. Jimenez, and M. Valencia, Dec. 2010, "High radix implementation of Montgomery multipliers with CSA," in Proc. Int. Conf. Microelec- tron., pp. 315–318.

[8] P. Amberg, N. Pinckney, and D. M. Harris, Oct. 2008, "Parallel high-radix Montgomery multipliers," in Proc. 42nd Asilomar Conf. Signals, Syst., Comput., pp. 772–776.

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

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

[11] Deepa Jose, Chitra M, P. Nirmal Kumar " Reliability Improvement Of Partitioned VLSI Systems For Fault Tolerance" in International Journal of Applied Engineering Research (IJAER) , vol 10 ,issue 7, pp. 18151-18165, 2015