African Journal of Basic & Applied Sciences 9 (5): 272-278, 2017
ISSN 2079-2034
© IDOSI Publications, 2017
DOI: 10.5829/idosi.ajbas.2017.272.278

Efficient VLSI Architecture for Montgomery Modular Multiplier

[1]S. Meenakshi and [2]M. Jagadeeswari

[1]PG-Scholar / Department of ECE, Sri Ramakrishna Engineering College, Coimbatore, India
[2]Professor & Head / Department of ECE, Sri Ramakrishna Engineering College, Coimbatore, India

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 multipliers.

Key words: Configurable Carry Save Adder · Skip detector

INTRODUCTION

An area-efficient and fast modular multiplication algorithms comes from their application in public key cryptography. Montgomery multiplication algorithm is a most efficient algorithm which is mostly used for modular multiplication. Montgomery multiplication computes S=a*b mod n, for positive integers a, b and n. When there are a large number of multiplications to be done with the same modulus n and with a small number of multipliers, it reduces execution time. But the division operations takes more execution time which becomes major difficulty in execution time.

The Montgomery reduction is used to find remainder of a division. Bit-wise and block-wise approaches are the existing FPGA implementations of Montgomery multiplication. To overcome the problem of long carry chains, bit-wise and block-wise systems uses the carry save techniques. Montgomery multiplication is used to speed up encryption and decryption process.

Literature Survey: A literature survey is done for various papers to know the available techniques and their significance and limitations. It also includes the various supporting papers for the proposed technique and their advantages. Several approaches are based on carry-save addition were done to achieve a significant speedup of Montgomery Multiplication.

Techniques such as parallelization, high-radix algorithm and systolic array design can be combined with the CSA architecture which further enhances the performance of Montgomery multipliers but causes a large increase in hardware complexity and power/energy dissipation. Square-multiply exponentiation method and CRT (Chinese Remainder Theorem) technique are also used to perform modular multiplication [1]. S.R. Kuang, J.P. Wang, K.C. Chang and H.W Hsu proposed an energy-efficient high throughput Montgomery modular multipliers for RSA cryptosystems [2]. The architecture is capable of bypassing the carry-save addition and register write operations, resulting in less energy consumption and higher throughput. In addition, a modified Barrel Register Full Adder (BRFA) significantly reduce the energy consumption of storage elements and also enhance the throughput of Montgomery modular multipliers. Fully systolic array architecture presents a high-radix implementation with carry propagation between the Processing Elements [3]. The parallel implementation has multipliers blocks in parallel with the Processing Elements and it provides a pipelining. Both architectures

Corresponding Author: S. Meenakshi, PG-Scholar / Department of ECE, Sri Ramakrishna Engineering College, Coimbatore, India.

have enhanced performance and speed improvement but it has larger area leading hardware complexity. Parallel implementation and partitioning for the Modular Multiplication are also done [4]. G. Perin, D. G. Mesquita, F. L. Herrmann and J. B. Martins proposed an Montgomery modular multiplication on reconfigurable hardware based Fully systolic array vs parallel implementation [5]. The modular multiplication is employed in modular exponentiation processes, which is the most important operation of some public-key cryptographic algorithms and the most popular of them is the RSA encryption scheme.

From the literature works, it is evident that the previous works have several disadvantages. Techniques such as parallelization, high-radix algorithm and systolic array design can be combined with the CSA architecture which further enhances the performance of Montgomery multipliers but causes a large increase in hardware complexity and power/energy dissipation.

Conventional Montgomery Modular Multiplier: In the conventional structure of Semi Carry Save Montgomery Modular Multiplier, one design uses the two carry save adders whereas the other design uses only one carry save adder with registers.

Fig. 3.1 shows the Block diagram of Semi Carry Save Montgomery Modular (SCSMM) Multiplier 1. In the Semi Carry Save Montgomery Modular Multiplier, the input and Output operands (i.e., A, B, N and Shift Carry (SC) and Shift Sum(SS)) are represented in binary and since to avoid carry propagation the intermediate results are represented in carry save format.

Carry Save Adders are used to avoid Long Carry Propagation to execute addition stages in Montgomery algorithm [6]. The parallel structure of this adders shows that shallow combinational logic depth is assured independent of the target technology on which Carry Save Adders are implemented. Hence this suitable for long integer arithmetic in RSA implementation [7].

Each Modular Multiplier perform Montgomery Multiplication in k or k+1 or k+2 clock cycles where k denotes operand bit length. To avoid the final comparison and subtraction the number of iterations may change from k to k+2 depending on the design of Montgomery Modular Multiplier. The select signals of Multiplexers M1 and M2 are generated by control part.

Fig 3.2 Block diagram of Semi Carry Save Montgomery Modular (SCSMM) multiplier 2. In the Semi Carry Save Montgomery Modular Multiplier design 2, the precomputation takes place by pre computing D = B + N and it uses one level CSA architecture. The precomputation takes place to reduce the critical path delay of Semi Carry Save based Modular Multiplier. The carry propagation addition operation is performed by one level CSA architecture until the shift Carry (SC) becomes zero. The pre-computation of A_i and q_i also takes place which is used to select the input operand 0, N, B and D in the Multiplexer M3. Many extra clock Cycles are required to perform three-input carry-save addition through one level CSA architecture since it is necessary in every Modular Multiplication.

Extra clock cycles for format conversion possibly lower the performance of SCS-based multipliers. To further enhance the performance of the SCS-based

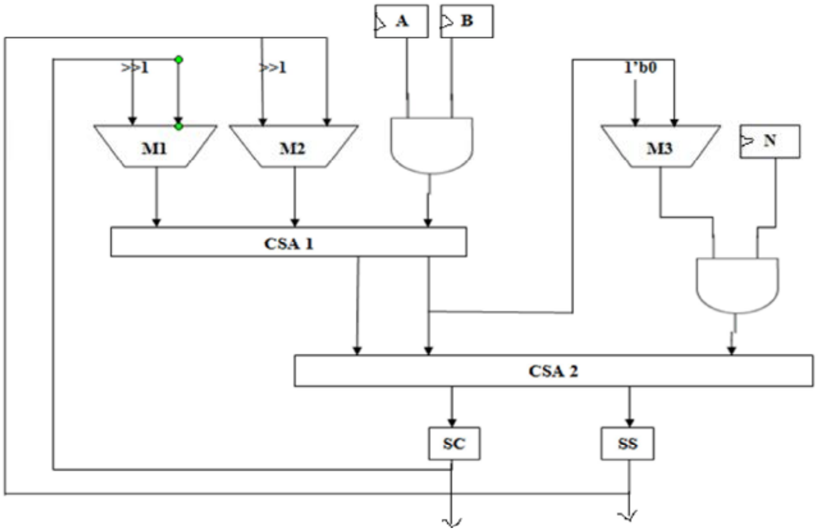

Fig. 3.1: Block diagram of Semi Carry Save Montgomery Modular multiplier 1

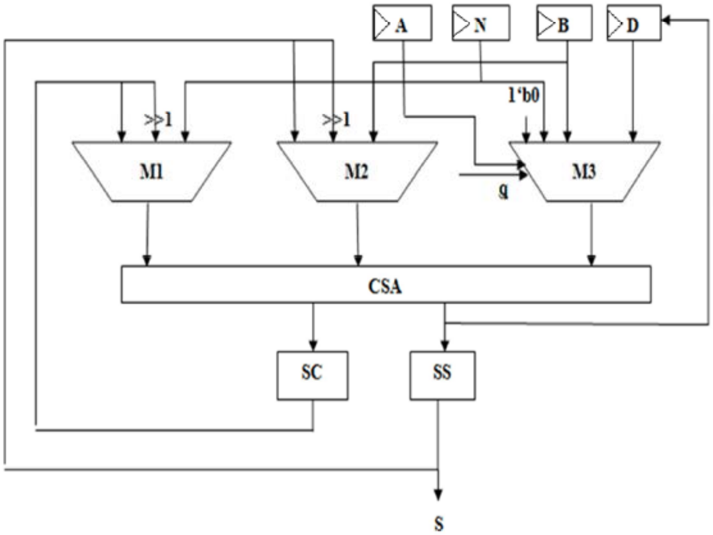

Fig. 3.2: Block diagram of Semi Carry Save Montgomery Modular multiplier 2

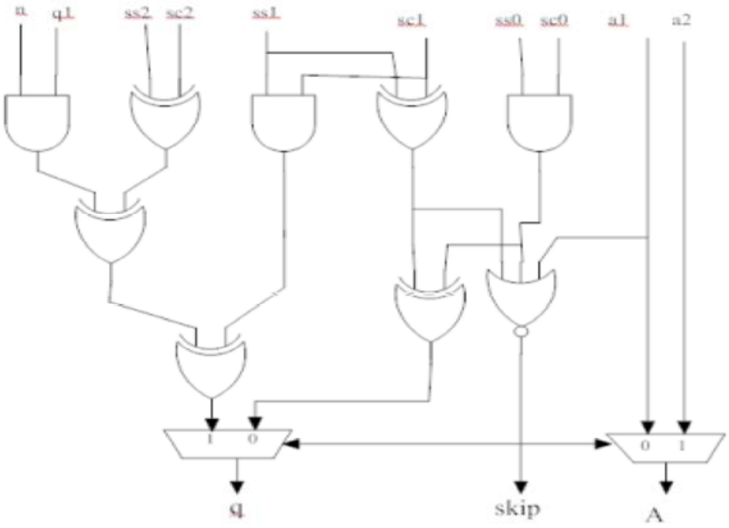

Fig. 4.1: Structure of Skip Detector

multiplier, both the critical path delay and clock cycles for completing one multiplication must be reduced while maintaining the low hardware complexity.

Proposed Montgomery Modular Multiplier: The proposed Montgomery multiplication aims at enhancing the performance of CSA-based Montgomery multiplier while maintaining low hardware complexity. The extra clock cycles for operand pre computation and format conversion can be hidden and high throughput can be obtained. The proposed algorithm and hardware architecture have the following several advantages and novel contributions over previous designs. In addition, the drawback of more clock cycles for completing one multiplication is also improved while maintaining the advantages of short critical path delay and low hardware complexity.

Skip Detector: A skip Detector is designed which is used to detect and skip unnecessary addition operations (i.e) unnecessary iteration, thereby maintaining short critical path delay. The skip detector is composed of five XOR gates, three AND gates and one NOR gate.

Skip detector is developed to generate skip, q and A in the iteration. The structure of skip detector is shown in Fig. 4.1.

Skip detector is used to avoid unnecessary iterations while performing modular multiplication.

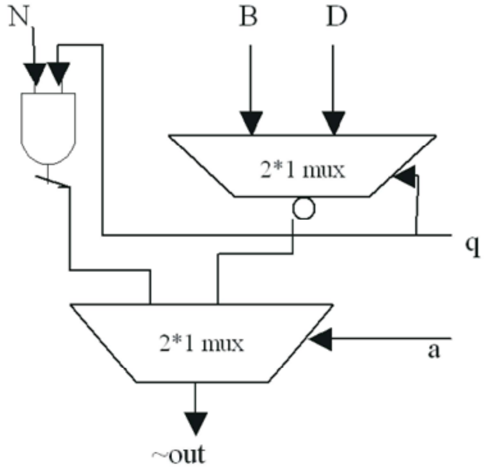

Fig. 4.2: Structure of Modified Mux design

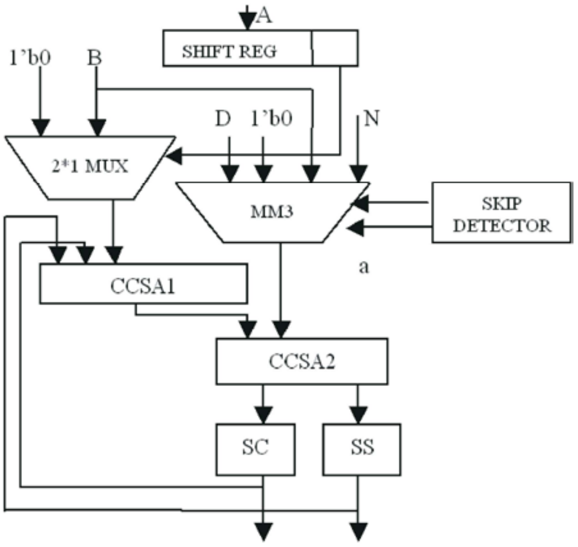

Fig. 4.3: Proposed Montgomery modular multiplier design 1

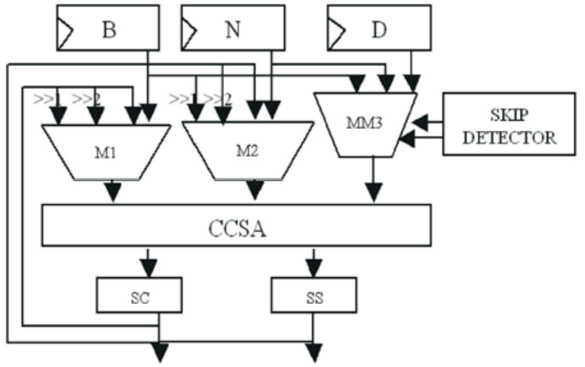

Fig. 4.4: Proposed Montgomery modular multiplier design 2

Modified MUX Design: The Modified MUX design consist two 2*1 multiplexers which is shown Fig 4.2.

The select input to multiplexers q and a is provided by means of skip detector. The 4-to-1 multiplexer M3 modified into a simplified multiplier MM3 because one of its inputs is zero, where ~denotes the INVERT operation.

Proposed Montgomery Modular Multiplierdesign1: In order to reduce the critical path delay, a new Semi Carry Save (SCS) based Modular Multiplication has been proposed.

A Configurable Carry Save Adder (CCSA) which uses two serial half adders has been designed to reduce the extra clock cycles which is used for operand pre-computation. The proposed Montgomery modular multiplier design 1 is shown Fig. 4.3.

The proposed design uses two-level Configurable CSA architecture to perform the carry-save addition. The Modified Mux design and a skip detector is used to enhance the speed by maintaining short critical path delay.

Proposed Montgomery Modular Multiplierdesign 2: The proposed Montgomery modular multiplier design 2 is shown Fig. 4.4.

The hardware architecture of proposed Montgomery multiplication consists of one level CCSA architecture, two 4-to-1 multiplexers (i.e., M1 and M2), one simplified multiplier, one skip detector.

Hence the proposed design results in short critical path delay and area. The performance of Montgomery modular multiplication is improved by the use of two serial half adders and skip detector. Significant performance gains achievable through proposed design.

Simulation: In this chapter the simulation results of conventional semi carry save Montgomery multiplication and proposed Montgomery modular multiplication are discussed. The various designs are coded in Verilog and simulation is carried out using the ISIM simulator and the synthesis is done in XILINX ISE Design suite 13.2. Based upon the obtained results of modular multiplication, comparison have been made on following parameters like area power and delay.

Simulation Result of Conventional Scsmm1: The simulation result of conventional scsmm1is shown in Fig. 5.1

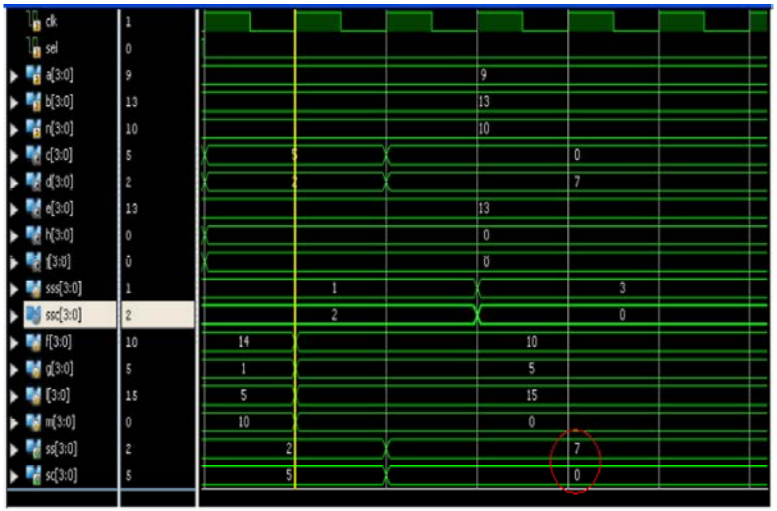
clk 1
sel 0
a[3:0] 9
b[3:0] 13
n[3:0] 10
c[3:0] 5
d[3:0] 2
e[3:0] 13
h[3:0] 0
i[3:0] 0
sss[3:0] 1
ssc[3:0] 2
f[3:0] 10
g[3:0] 5
l[3:0] 15
m[3:0] 0
ss[3:0] 2
sc[3:0] 5

Fig. 5.1: Simulation of conventional SCSMM1

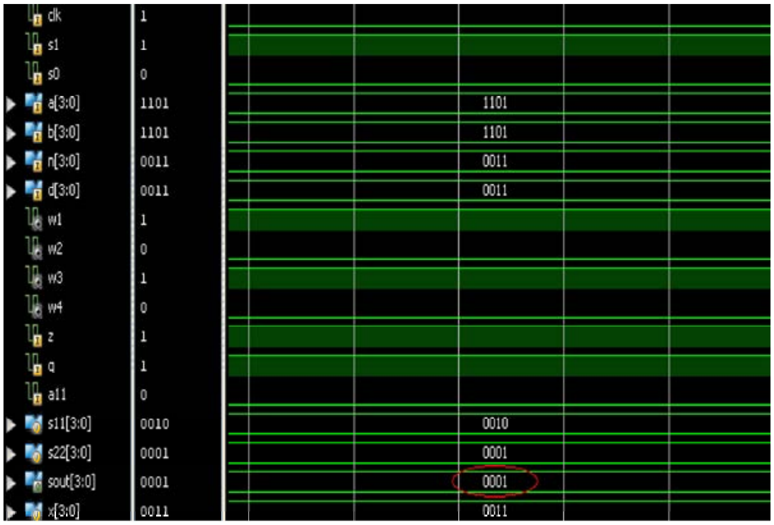
clk 1
s1 1
s0 0
a[3:0] 1101
b[3:0] 1101
n[3:0] 0011
d[3:0] 0011
w1 1
w2 0
w3 1
w4 0
z 1
q 1
a11 0
s11[3:0] 0010
s22[3:0] 0001
sout[3:0] 0001
x[3:0] 0011

Fig. 5.2: Simulation of conventional SCSMM2

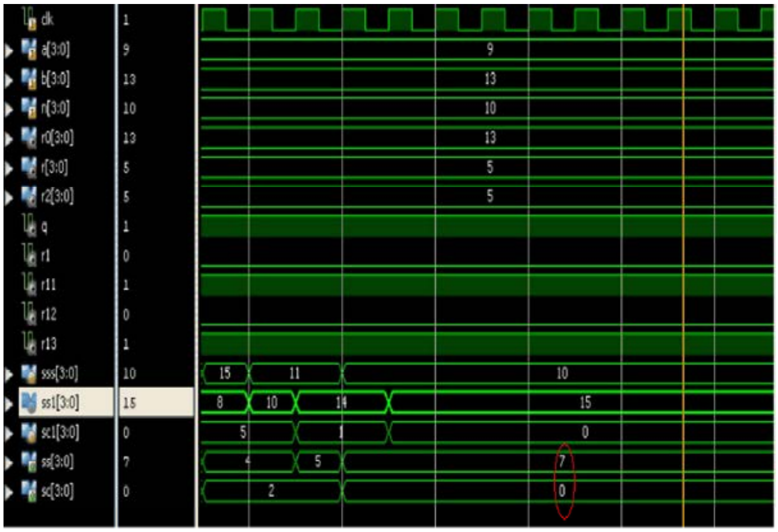
clk 1
a[3:0] 9
b[3:0] 13
n[3:0] 10
r0[3:0] 13
r[3:0] 5
r2[3:0] 5
q 1
r1 0
r11 1
r12 0
r13 1
sss[3:0] 10
ss1[3:0] 15
sc1[3:0] 0
ss[3:0] 7
sc[3:0] 0

Fig. 5.3: Simulation of proposed design 1

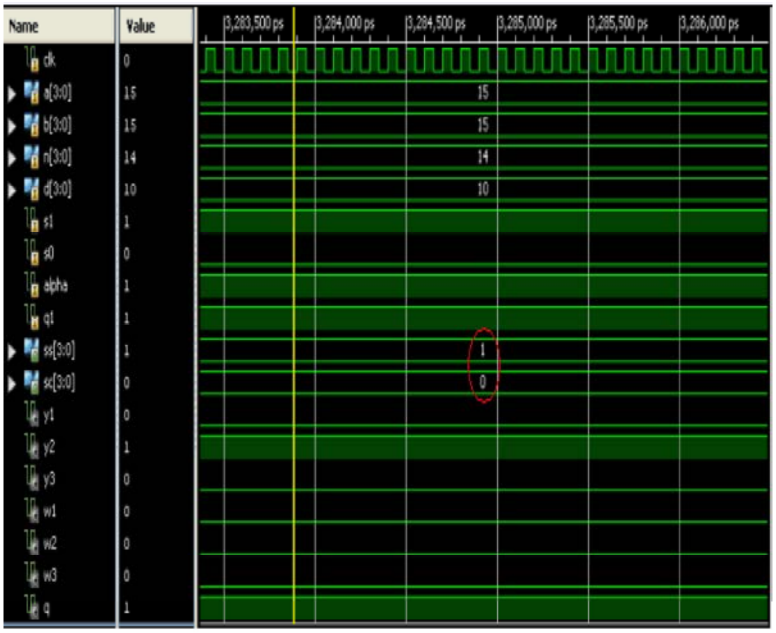

Fig. 5.4: Simulation of proposed design 2

Table I: Comparison of Modular Multipliers

Parameters	Scsmm1	Scsmm2	Proposed Design 1	Proposed Design 2
Area				
No of Slice Luts	22	8	6	6
No of Occupied Slice	17	4	4	4
No of IOs	18	16	13	13
Memory Usage (kb)	137072	137072	137072	137072
Delay (ns)	5.753	4.040	3.492	1.777
Power (W)	0.034	0.034	0.034	0.034

Simulation Result of Conventional Scsmm 2: The simulation result of conventional scsmm2 is shown in Fig.5.2

Simulation Result of Proposed Design 1: The simulation result of proposed design 1 is shown in Fig. 5.3.

Simulation Result of Proposed Design 2: The simulation result of proposed design 2 is shown in Fig. 5.4

Table I shows the analysis of various parameters such as AREA, POWER and DELAY of both conventional and proposed Multipliers.

CONCLUSION

The conventional and proposed Montgomery multiplication is analyzed. The proposed multiplier used two-level and one-level CCSA architecture and skip detector which skipped the unnecessary carry-save addition operations to largely reduce the critical path delay and required clock cycles for completing one modular multiplication operation. The Design entry is done through Verilog coding in Xilinx environment and it is analyzed that the proposed design gives better results when compared to conventional design. The experiment results showed that the AREA (in terms of number of slice LUTs 6 out of 1920 and in terms of number of occupied slice 4 out of 960) and DELAY obtained by proposed design 1 is 3.429ns whereas the delay obtained by proposed design 2 is 1.777ns which achieves higher performance and significant area–time product improvement when compared with conventional Montgomery modular multiplication designs.

ACKNOWLEDGMENT

I also extend my thanks to my guide Dr. M.Jagadeeswari, M.E., Ph.D., Professor & Head, Department of Electronics and Communication Engineering, who had been a source of inspiration, encouragement and cooperation in carrying out the project work and for her timely guidance in the conduct of my project work.

REFERENCES

1. Aoki, T., N. Homma, A. Miyamoto and A. Satoh, 2011. Systematic design of RSA processors based on high-radix Montgomery multipliers, IEEE Trans. Very
2. Al Shboul, R.M., V.P. Shirochin and H. Zhengbing, 2007. An efficient architecture of 1024-bits cryptoprocessor for RSA cryptosystem based on modified Montgomery's algorithm, in Proc. 4th IEEE Int. Workshop Intell. Data Acquisition Adv. Comput. Syst., pp: 643-646.
3. Choi, J.R., W.S. Kang and Y.S. Kim, 2000. Asynchronous implementation of 1024-bit modular processor for RSA cryptosystem, in Proc. 2nd IEEE Asia-Pacific Conf. ASIC, Aug. pp: 187-190.
4. Han, J., W. Huang, S. Wang, Z. Yu and X. Zeng, 2013. Parallelization of radix-2 Montgomery multiplication on multi core platform, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 21(12): 2325-2330.
5. Herrmann, F.L., D.G. Mesquita, J.B. Martins and G. Perin, 2010. Montgomery modular multiplication on reconfigurable hardware: Fully systolic array vs parallel implementation, in Proc. 6th Southern Program. Logic Conf., pp: 61-66.
6. Lin, W.C., M.D. Shieh, J.H. Ye and S.H. Wang, 2012. Fast scalable radix-4 Montgomery modular multiplier, in Proc. IEEE Int. Symp. Circuits Syst., pp: 3049-3052.
7. Li, Z., L. Yang, S.W. Zhang and Y.Y. Zhang, 2007. An efficient CSA architecture for Montgomery modular multiplication, Microprocessors Microsyst., 31(7): 456-459.