Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 10 (2017) pp. 3085-3094

http://www.ripublication.com

FCS Based Montgomery Modular Multiplication

[1]Bhagyasree Kokkonda and Dr. Tarun Kumar Juluru[2]

[1] PG Scholar, Electronics Design Technology, S R Engineering College Warangal, Telangana, India
[2] Professor , Department of Electronics and Communication Engineering S R Engineering College, Warangal, Telangana, India

Abstract

Security plays an important role in the present generation. It is considered in three matters i.e., confidentiality, integrity & availability. All this will come with an efficient cryptographic algorithm which is achieved by repeating modular multiplications on large integers. To increase the speed by encryption decryption process, high speed Montgomery modular multiplication algorithms, hardware architectures employs carry save addition to avoid the carry propagation at the addition operation of the add-shift loop. We proposed an energy-efficient algorithm and its corresponding architecture are not only to reduce the energy consumption but also further enhance the throughput of Montgomery modular multipliers.

This architecture which is capable of bypassing the superfluous carry-save addition and register write operations, which leads to less energy consumption. In addition we modified the barrel register full adder (BRFA) so that the gated clock design is applied significantly to reduce the energy consumption of storage elements in BRFA.

Keywords: Carry-save adder, Gated clock delay, Montgomery modular multiplier, RIVEST, SHAMIR, and ADLEMAN(RSA) cryptosystem.

1. INTRODUCTION

Increase in the use of Internet and electronic era has brought secrecy in electronic communication. More secrecy i.e., information are transferred and saved every day so security is the major issue.

Encryption is the standard method for making a private communication. If someone wants to send a confidential message they need to encrypt the message before transmitting to the receiver. The receiver only can decode the encrypted message.

Besides the receiver, anyone who sees the message can only see the encoded message. Even if they try to decode the message they may not get the original message, definitely, there will be a loss of information.

Many use public key cryptography in security operations. The size of the modulus is the maximum of 1024 bits in the process of modular exponentiation. More data throughput, without the speed hardware, is highly impossible. Other than this providing secrecy is also essential because of less battery power for electronic devices like notebook computers, smartphones etc.

Kuang et al suggested an FCS based multiplier which is operated in 4 to 2 level CSA architecture .are used in emancipation energy and to increase throughput.

The only problem of FCS MMM42 Multiplier is a high area, complexity and longer delays which is not suitable for portable systems.

LITERATURE SURVEY

[1]In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman proposed a cryptographic algorithm, which was very important to replace the less secure National Bureau of Standards (NBS) algorithm. RSA proposed a public-key cryptosystem, as well as digital signatures. RSA suggested two significant ideas:

1.Public-key encryption. In this, the recipients do not require a courier to deliver the keys before transmitting the original signal. In RSA, the person who has the exact decryption key can decipher/unlock an encrypted message since the encryption keys are public, while the decryption keys are not. Encryption and decryption keys are different for different people. The keys are prepared in such a way that the decryption key cannot easily deduce from the public encryption key.

2. Digital signatures. In this also the message is transmitted from the sender to the receiver but at the receiver side, the receiver has to verify that this message is originally sent by the sender or not. This can be done by using the sender's decryption key afterward the signature can be verified by using the public encryption key. Therefore, Signatures cannot be duplicated.

[2] The study of ciphers is known as Cryptography which is very much secure and protects the data from the unauthorized users. In brief, cryptography is the process of converting the plaintext into ciphertext and vice versa. These can be executed by using secret keys and a cryptographic algorithm. The message that a sender is going to send is known as plaintext and the message that is sent through the channel is known as ciphertext. In modern cryptography different secret keys are used for encryption and decryption known as asymmetric key cryptography.

2 RELATED WORK

2.1 ARCHITECTURE OF EXISTING SYSTEM

The MMM42 multiplier block diagram of existing system Figure. 1. Furthermore MBRFA and LU, the important element is the 4-to-2 CSA architecture. The CSA architecture can be integrated with other techniques and structures to increase the performance of MM multipliers. Therefore, these designs possibly cause more power consumption which also increases the hardware complexity. So we have proposed a new Montgomery algorithm which reduces the energy consumption.

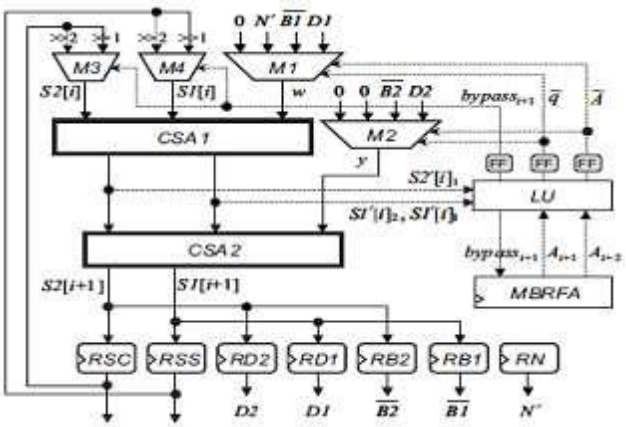

Figure 1: Block diagram of existing system MMM42 Multiplier

In the subsequent chapters, we will submit a new Montgomery algorithm to bypass the redundant operations without growing the critical path delay in Montgomery multipliers using 4-to-2 RCA architecture is proposed to reduce the energy/power utilization and amplify the throughput via bypassing the remaining operation.

Algorithm MMM42: 4-to-2 CSA Modified Montgomery Multiplication

Inputs:1, $A2$, $B1$, $B2$, $N(modulus)$

Outputs:1$[K]$, $S2[K]$

1.($D1$,$D2$) = $B1$ + $B2$ + N +0;

2. $S1[0] = 0$; $S2[0] = 0$;

3. $for\ i = 0\ to\ k - 1$ {0

4. qi = ($S1[i]0$ + $S2[i]0$ + Ai × ($B10$ + $B20$

)) 2;

5.($Ai = 0\ and\ qi = 0$)

6.($S1[i + 1], S2[i + 1]) = (S1[i]+ S2[i]+ 0 + 0)⁄2$

7.$else\ (Ai = 0\ and\ qi = 0)$

8. ($S1[i + 1], S2[i +1]) = (S1[i] + S2[i] + N + 0)⁄2$;

9.$else\ if\ (Ai = 1\ and\ qi = 0)$

10.($S1[i + 1], S2[i + 1]) = (S1[i] + S2[i] + B1 + B2)⁄2$

11.$else\ (Ai = 1\ and\ qi = 1)$

12. ($S1[i + 1], S2[i + 1]) = (S1[i] + S2[i] + D1 + D2)⁄2$

13.0}

14. Return $S1[K], S2[K]$

As shown in Algorithm MMM42, steps 1 and 2 for producing the carry values (B1, B2) and (D1, D2) are first performed to easily realize the subsequent quotient look ahead. Because the qi+1, Ai+1, qi+2, and Ai+2 must be generated at the ith iteration, the iterative index i of Montgomery modular multiplication will start from −1 instead of 0 and the corresponding initial values of ˜q and ˜A must be set to 0.

Additionally, the original for loop in Algorithm MMM42 is replaced with the while loop to bypass some superfluous iterations when bypass i +1 = 1.Note that the ending number of iterations in Algorithm MMM42 is changed to k + 2 instead of k−1 due to the following reasons. First, the convergence range of S in Algorithm MM falls in the range of [0, 2N), thus an additional operation S = S − N is required to keep the range of output S in [0, N) if S ≥ N. We employ the notion of Walter's approach [24] to remove the time-consuming subtraction and maintain the range of input operands A, B, and output S within [0, 2N) through increasing two extra iterations and extending (A1, A2) and (B1, B2) to k + 2 bits .Because (A1, A2) and (B1, B2) are unsigned numbers, two dummy zeros are directly inserted in front of the MSB and the carry values (B1, B2) = 2B1 +2B2 is calculated at the beginning of Algorithm MMM42.

Therefore, an extra iteration for computing division by two is necessary to ensure the correctness of Montgomery modular multiplication. In this manner, the output (S1, S2) can be utilized as an input in the consecutive modular multiplication without the additional Subtraction while carrying out the modular exponentiation. In the while loop, steps 6– 13 will be performed in a four-to-two RCA architecture with two 4-to-1 multiplexers. Computations of Ai+1 and Ai+2 in step 14 can be executed in parallel with the four-to-two RCA architecture. In addition, computations of qi+1, qi+2, and bypass i+1 in step 14 can be carried out after S1_[i]1, S2_[i]1, and S1_[i]2 have been produced by RCA1 (i.e., the upper FAs of the four-to-two RCAs architecture) Subsequently, steps 15–21 except step 17 can be performed after step 14 has been

completed.

In the present section, the basic architecture of MMM42 algorithm is represented as the MMM42 multiplier. Then, to the MMM42 multiplier the gated clk., design technique is applied to decrease the power/energy consumption.

2.2 Modified Barrel Register Full Adder (MBRFA) and Look-Ahead Unit (LU)

In the above algorithm MMM42, ˜q and ˜A be required to generate and stores the Ith iteration value accordingly to bypass i+1 signal in order to select the exact input signal operands of 4-to-2 carry save adder at the succeeding clk., cycle bit. Therefore, Ai+1, Ai+2 both these two signals are required to decrease ˜A. However, the operands(values) Ai+1 and Ai+2 are generated by modifying the BRFA which is shown in figure 1.The restructured BRFA (represented as MBRFA) illustrated in Fig. 2 uses two shift registers RA1 and RA2 along with two full adders to produce Ai+1 and Ai+2 at the same clock cycle.

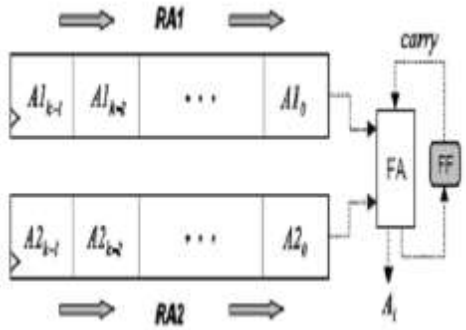

Figure 2: BRFA

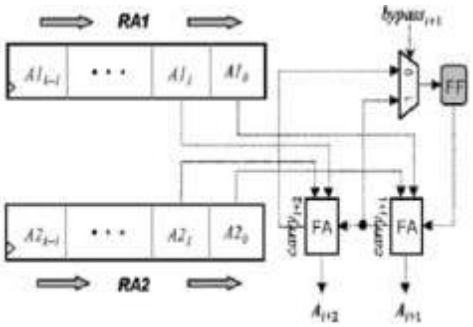

Figure 3: MBRFA

LU rendered in Figure. 4 is evolved to obtain the bypass i+1 signal, ˜q, and ˜A after obtaining the operands Ai+1 and Ai+2. The LU is embedded with an XOR gate, a NOR gate, and two 2-to-1 multiplexers. It generates the qi+1, qi+2, and bypass i+1

signal simultaneously.

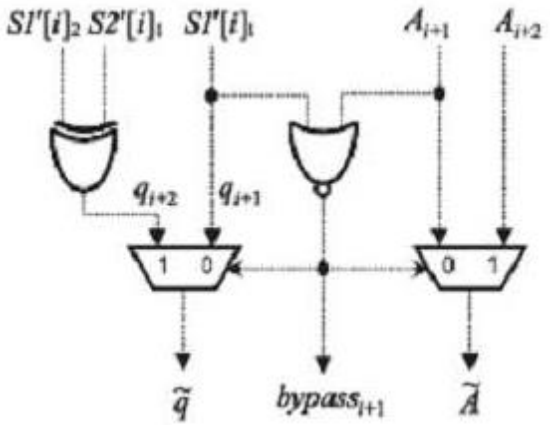

Figure 4: Look-Ahead Unit (LU)

Eventually, ˜q and ˜A are selected. The bypass i+1 signal should give to MBRFA to get the carry operand value that must be stored in FF where RA1 and RA2 perform the right shift operation.RA1 and RA2 perform right shift operation by two-bit positions when the bypass i+1 = 1 and carry i+2 are stored to Flip-flop. Or else the RA1 and RA2performs right shift operation by one-bit position when the carry i+1 is stored to Flip Flop.

3. PROPOSED MODELLING

The block diagram of proposed MMM42 Montgomery multiplier is shown in Figure5. Besides LU and MBRFA, the 4 to 2 RCA architecture is the important component. Firstly, in step1 and step2, the four-to-two (4 to 2) RCA operand values are computed by adjusting the ˜A, ˜q. The operands (B1, B2) are given to the RB1 and RB2 registers, and the operands (D1, D2) are given to the RD1 and RD2 registers, simultaneously. In the iteration process I for When carrying out the computation of iteration i for −1 ≤ i ≤ k + 2, the 2- to- 2 through the M1 and M2 multiplexers, correct w and y signals are selected which uses ˜A and ˜q obtained at the previous clock cycle.

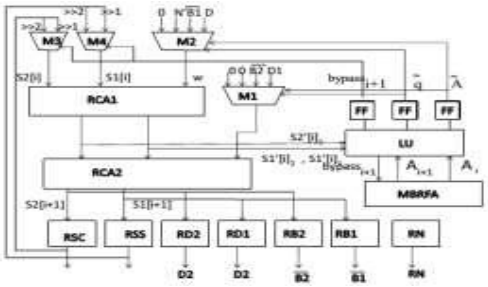

Figure 5: Block diagram of proposed MMM42 Multiplier.

The results S1[i+1] and S2[i+1] of 4-to-2 RCA algorithm must be divided by two as shown in the steps 6–13 of Algorithm. While bypass i+1 = 1, therefore, the outputs of 4-to-2 RCA architecture is divided by 4 (i.e., right-shifted by two-bit positions). One possible method to perform the extra right-shift one-bit operation (i.e., step 17 of Algorithm MMM42) is inserting two extra 2-to-1 multiplexers M3 and M4 with bypass i+1 as the select signal to the front of RSCj and RSSj, respectively. If bypass i+1 = 1, M3 and M4 will select S1[i+1] j+1 and S2[i+1] j+1, respectively. Otherwise, S1[i+1] j and S2[i+1] j will be selected through M3 and M4. Instead of the above method, we store the bypass i+1 signal to FF and insert multiplexers M3 and M4 at the front of RCA1 as shown in Figure. 5.

Therefore, the right-shift one-bit or two-bit operation will be performed at the next clock cycle according to the bypass i+1 signal stored in FF. The glitches of bypass i+1 signal, perhaps bring purposeless dynamic power/energy consumption will be blocked/avoided by Flip Flop and not get passed through M3, M4, and RCA. The critical path delay is not changed even if M3 and M4 are connected in parallel with M1.

3.1 FCS-Based Montgomery Multiplication

In order to lessen the format conversion, This Montgomery multiplication enables A, B, and S in the carry save representations (AS, AC), (BS, BC), including (SS, SC), individually. C McIvor et al. submitted the two FCS based Montgomery multipliers which are symbolized as FCS-MM-1 and FCS-MM-2 multipliers, denoted as 5-2(five to two) (3-level) and another is 4-2(four-to-two) (2-level) CSA architecture, respectively. The barrel register full adder (BRFA) embrace of 2 shift registers to store AS and AC values, a flip-flop (FF). and a full adder (FA), along with this the FCS-MM-2 multiplier suggested is added with BS, BC, and N into DS and DC at the first of the every Montgomery multiplier. The deepness of the CSA tree is decreased from 3 to 2 levels. However, the FCS-MM-2 (multiplier) requires 2 extra 4-to-1(four-to-one) multiplexers represented by Ai and qi along with the data storing elements DS, DC in order to decrease the 1 level in CSA. Hence, in FCS-MM-2 technique the critical path is reduced to a greater extent but the drawback is it has increased the area when compared to the FCS-MM-1 multiplier.

3.2 Gated Clock Design Technique

The energy consumption or the power can be minimized by using gated clock design will also reduce the power consumption of MMM42 multiplier since many registers were used to store the inputs in the multiplier in order to get the better results in the output, the registersRB1, RB2, RD1, RD2, and RN in Figure 5., takes new value while performing the Montgomery multiplication operations.

4 RESULTS AND DISCUSSIONS

The written Verilog HDL Modules have successfully simulated and verified using Modelsim III 6.4b and synthesized using Xilinx ISE 14.7

Simulation Result:

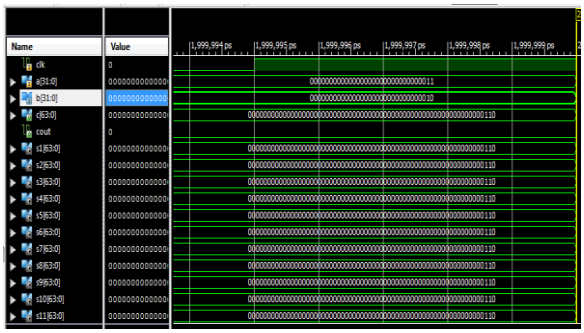

Figure 6 Simulation Result

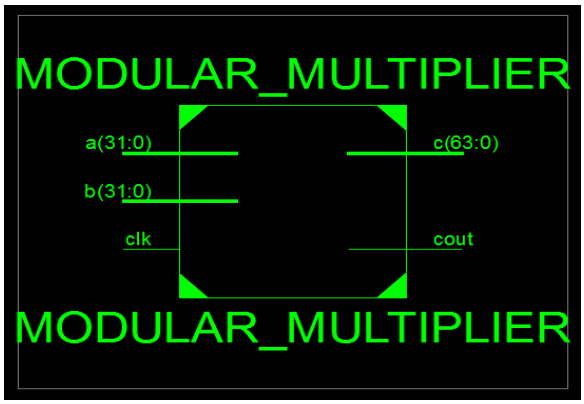

Figure 7: RTL Schematic

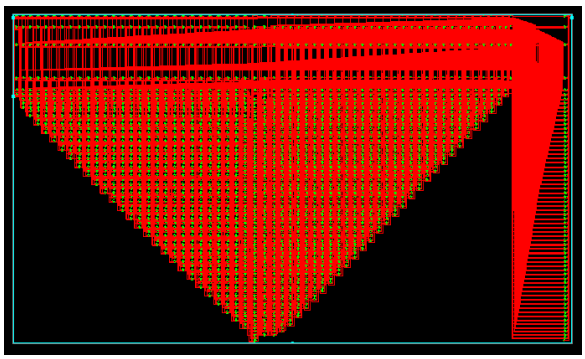

Figure 8 Technology Schematic

Timing Report:

```
Offset:             4.040ns (Levels of Logic = 1)
  Source:           m8/data_26 (FF)
  Destination:      y1<25> (PAD)
  Source Clock:     clk rising

  Data Path: m8/data_26 to y1<25>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q             1   0.514   0.357  m8/data_26 (m8/data_26)
     OBUF:I->O                3.169          y1_25_OBUF (y1<25>)
    ----------------------------------------
    Total                     4.040ns (3.683ns logic, 0.357ns route)
                                      (91.2% logic, 8.8% route)
```

Figure 9 Technology Schematic

5. CONCLUSION

High energy/power consumption and more number of registers were introduced into the high-speed Montgomery Modular multipliers, which amplify the speed in encryption/decryption procedure by maintaining all inputs and outputs of the MM Multiplier in a redundant carry-save format. This paper provided an efficient algorithm and its architecture which reduces the energy/power consumption and increase the throughput of Montgomery Modular multipliers respectively. Consequently, the BRFA structure is modified and the gated clock design technique is used to decrease the energy consumption of the Montgomery Modular Multiplier. In future, we will try to heighten the occurring probability of superfluous operation bypassing to further reduce the energy consumption and increase the throughput of modular multiplication.

REFERENCES

[1] R. L. Rivest, A. Shamir and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” Communication ACM, Vol. 21, No. 2, 1978, pp. 120-126.

[2] P. L. Montgomery, “Modular multiplication without trial division,” Mathematics of Computation, Vol. 44, no. 170, pp. 519–521, Apr. 1985

[3] C.K.Koc, T.Acar, and B.Kaliski, "Analyzing and comparing Montgomery Multiplication algorithms", IEEE Micro, Vol. 16, no. 3, pp. 26–33, June 1996.

[4] Y. S. Kim, W. S. Kang, and J. R. Choi, "Implementation of 1024-bit modular processor for RSA cryptosystem", in Process IEEE Asia-Pacific Conference, Aug. 2000, pp. 187–190.

[5] V. Bunimov, M. Schimmler, and B. Tolg, "A complexity-effective version of

Montgomery's algorithm", in Process Workshop on Complexity Effective Designs, May 2002,pp. 1–7.

[6] A. Cilardo, A. Mazzeo, N. Mazzocca, and L. Romano,"A novel unified architecture for public-key cryptography", in Process Design, Autom. Test Eur. Conference Exhibit., Mar. 2005,pp. 52–57.

[7] Z. B. Hu, R. M. A. Shboul, and V. P. Shirochin, "An efficient architecture of 1024-bits Crypto processor for RSA cryptosystem based on modified Montgomery's algorithm", in Process 4th IEEE Int. Workshop Intell Data Acquisit. Advanced Computer Systems, Sep. 2007, pp. 643–646.

[8] C. McIvor, M. McLoone, and J. V. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques", IEE Process-Computer Digital Technology,Vol. 151, no. 6, pp. 402–408, Nov. 2004.

[9] K. Manochehri and S. Pourmozafari, "Fast Montgomery modular multiplication by pipelined CSA architecture", in Proc. IEEE International Conference Microelectron, Dec. 2004,pp.144-147.

Authors

Bhagyasree Kokkonda is pursuing Master of Technology in Electronic Design Technology in S R Engineering College, Warangal, Telangana. She has completed her B.Tech from Kamala Institute of Technology & Science, Huzurabad, Karimnagar, Telangana in 2014. Her areas of interests are VLSI, Digital Signal Processing, and Embedded Systems.

Dr. J. Tarun Kumar working as Professor , Department of Electronics and Communication Engineering, S R Engineering College, Warangal. He has 18 years of teaching experience. He has obtained B.Tech (ECE) Degree from MIET, Nagpur, India, M.Tech Degree from Kakatiya Institute of Technology & Science, Warangal, Andhra Pradesh, India and Ph. D. from JNTUH, Andhra Pradesh. His research areas include VLSI, Signal Processing, and Wireless Communications.