ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 15, March 2017

High performance Montgomery Modular multiplier using Carry Save Adder and Kogge Stone Adder

Greshma Eldhose
Dept. of Electronics and Communication
Christ Knowledge City
Ernakulam, Kerala
greshmaeldhose@gmail.com

Betcy K Joy
Dept. of Electronics and Communication
Christ Knowledge City
Ernakulam, kerala
k.betcyjoy@gmail.com

***Abstract*— This paper propose a High Performance Montgomery Modular Multiplier using Carry Save Adder (CSA) and Kogge Stone adder (KSA). It is mainly used in public cryptography. Kogge stone adder is a parallel prefix carry look ahead adder. It generates carry in O (logn) time and is widely considered as the fastest adder and is widely used in the industry for high performance arithmetic circuits. It can compute two inputs additions. To perform three input addition the Carry Save Adder is used. The Carry Save Adder (CSA) is used to avoid the large carry propagation.**

Keywords—Montgomery Modular Multiplier;Kogge Stone Adder; Carry Save Adder;

I. INTRODUCTION

Modular multiplication is a critical operation in the public key cryptosystem. This is because the modular exponentiation operation in the public key cryptosystem is obtained by using number of modular multiplication. Thus the performance of the public key cryptosystem depends on the modular multiplication. To make the modular multiplication faster a method called Montgomery modular multiplication is used. The earlier method to find out the modular multiplication is, by performing multiplication and then subtracting the modulus value in several times until the result is less than the value of modulus. Montgomery modular multiplication replace this operation by a series of shifting modular addition. Then the result will be S = A * B * R-1 mod N. where A ,B and N are the inputs. N is a k bit integer, R = 2k mod N.

The problem of long carry propagation in the Montgomery modular multiplication is an important issue. To avoid the long carry propagation of three operand addition in the iteration loop, the carry save adder can be used. i.e eliminate the propagation of carry by saving the carry in a register. The carry save adder is a multiple bit of full adder. It having the same delay as full adder because it produce the output in parallel. And it also allow high clock cycle.

Kogge Stone Adder (KSA) is a parallel prefix adder. It is widely in high performance arithmetic circuits because it considered as a fastest adder. It perform the addition by three steps: pre processing , carry look ahead , post processing.

In pre processing, it compute generate and propagate signal for each pair of input as:

$$pi = Xi \text{ xor } Yi$$

$$gi = Xi \text{ and } Yi$$

Carry look ahead network, it involve the computation of carries. It use group propagate and generate which is given by the equation,

$$Pi:j = pi:k+1 \text{ AND } pk:j$$

$$Gi:j = gi:k+1 \text{ or } (pi:k+1 \text{ and } gk:j)$$

The final step post processing is used to compute the sum bit as,

$$Si = pi \text{ xor } Ci-1$$

II. MONTGOMERY MODULAR MULTIPLIER USING CCSA

In Semi Carry Save Montgomery modular multiplication strategy, it first pre compute the value of D and then it goes to the iteration loop. The intermediate results are stored in carry save format. The fig 1 shows the block diagram of conventional semi carry save Montgomery modular multiplier:

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 15, March 2017

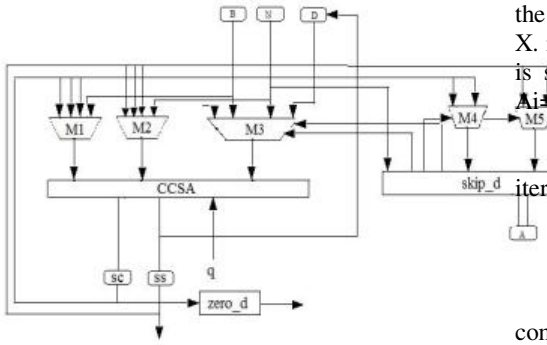

Fig .1. Montgomery modular multiplier

The CCSA indicate the configured carry save adder, i.e it select either one three operand carry save adder or two serial two operand carry save adder. To select these configuration a select input called q is given to the carry save adder.

In this , it first pre compute the value of D. To pre compute the value , it add B+N using CCSA and it result sum and carry. It repeat until the value of carry become zero. To check whether the value of carry equals to zero , zero_d circuit id used. This zero_d circuit can be easily implemented using nor operation.

The mux M3 select 0,N,B,D according to the select input Ai and qi.

Ai=0 and qi=0 ; X = 0

Ai=0 and qi=0 ; X =B

Ai=0 and qi=0 ; X =N

Ai=0 and qi=0 ; X = D

By skipping the unnecessary operations in the loop , we can reduce the number of clock cycle for completing one Montgomery modular multiplication. The complex computation in the loop is ,

$$(SS[i+1],SC[i+1])=(SS[i]+SC[i]+X)/2$$

Here , the value of X can be 0,B,N or D. It is depending upon the value of Ai and qi. If the value of Ai=0 and qi=0 then x=0. In this case, simply right shift the SS[i] and SC[i] to obtain the SS[i+1] and SC[i+1]. For this , a signal called skipi+1 is used to indicate that whether the (i+1)th iteration will be skipped or not. The equation of skipi+1is,

$$Skipi+1=\sim(Ai+1|qi+1|SS[i+1]0)$$

If the skipi+1is 0, then (i+1)th iteration cannot be skipped otherwise it can be skipped to the next iteration ie,(i+2). If it is skipped to the (i+2) th iteration ,then there is a need to know the value of Ai+2 and qi+2 to immediately select the value of X. i.e according to the value of skipi+1, the value of Ai and qi is selected. If skipi+1=0, Ai=Ai+1 and qi=qi+1. Otherwise Ai=Ai+2 and qi=qi+2.

It is easy to get the value of Ai+1 and Ai+2 in the i th iteration but in the case of qi+1 is obtained by the equation,

$$qi+1= (SS[i+1]0+SC[i+1]0+Ai+1*B0)\bmod 2.$$

The value of SS[i+1] and Sc[i+1] is available after the completion of (1). And also the critical path of Montgomery multiplier is increased.

To avoid this, the value of N,B, is modified to ensure that their least significant bits are equals to 0. i.e N is an odd number and it is used only when the value of qi=1. From this we can say that it propagate at least one carry. So modify the value of N as,

N_cap= N+1 if N1:0=11
N_cap= 3N if N1:0=01.

Also modify the value of B_cap as 8B to make the B2:0equals to0. Then the computation of qi+1 can by simplified eliminating Ai+1*B0. But require extra two clock cycle to correct the answer. Next add the two modified value of A and B to get D.
By using the modified the value of B ,N and D , the X2:0 become zero. So that it become easy to calculate SC[i+1]0 by SS[i]0&SC[i]0 and SS[i+1]0 by SS[i]0^SC[i]0. Then the equation of qi+1 ,

$$qi+1=((SS[i]0\&SC[i]0)\hat{}(SS[i]1\hat{}SC[i]1).$$

In the case of qi+2, it selected only if the value of Ai=qi=0 and SS[i+1]0=0 and SC[i+1]0=0. SC[i+2]0=SS[i]1&SC[i] 1then SS[i+2]0=SS[i]2^SC[i]^X2. X2 can be 0,N_cap2 ,0,D_cap2. i.e X2=qi^1'b1.

By using this , simplified the equation for skipi+1 as,

$$Skipi+1= \sim(Ai+1\&(SS[i]1\hat{}SC[i]1)\&(SS[i]0\hat{}SC[i]0)).$$

By using this , it easy to obtain skipi+1 in the i th iteration.

III. MONTGOMERY MULTIPLIER USING CSA AND KSA

Kogge stone Adder, which is widely used in fast arithmetic circuit can be used in this Montgomery multiplier to further improve the performance of Montgomery multiplier.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 15, March 2017

In the Montgomery multiplier the two operand operation is performed in two times, one is to pre compute the value of D and one for the format conversion from carry save format to the binary format. These operation can be done by using Kogge Stone Adder to perform it faster. [4] proposed a system in which the complex parallelism technique is used to involve the processing of Substitution Byte, Shift Row, Mix Column and Add Round Key. Using S- Box complex parallelism, the original text is converted into cipher text. From that, we have achieved a 96% energy efficiency in Complex Parallelism Encryption technique and recovering the delay 232 ns. The complex parallelism that merge with parallel mix column and the one task one processor techniques are used. In future, Complex Parallelism single loop technique is used for recovering the original message.

IV. RESULTS

A. *Montgomery Modular Multiplier Using CCSA*

The design is implemented in modelsim using verilog coding. The output waveform is shown below

The input of the Montgomery modular multiplier is A, B, rst, clk and N. In which A>0 and B<N. The values of A,B and N is selected in such a way they are prime numbers. N is a k bit modulus and odd number is widely used in practical applications.

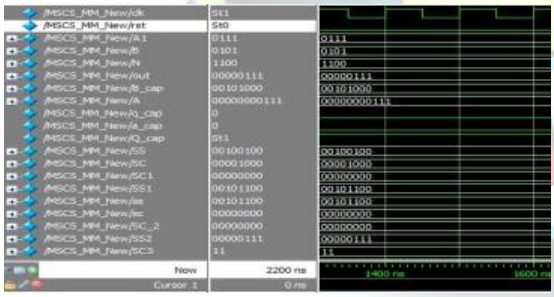

Fig .2. output wave form of MMM using CCSA

B. *Montgomery Modular Multiplier Using CSA and KSA*

The design is implemented in modelsim using verilog coding. The output waveform is shown below

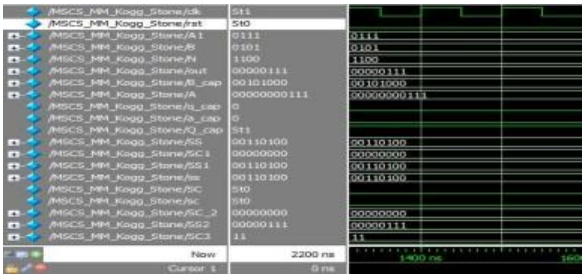

Fig .3. output waveform of MMM using CSA and KSA

V. CONCLUSION

This paper proposed a method to improve the critical path delay , number of clock cycle for the Montgomery modular multiplier using a configurable carry save adder. To further improve the performance of the Montgomery modular multiplier Kogge Stone Adder is used.

REFERENCE

[1] IANN-Rong Kung, Kun-Yi Wu, "low cost high performance VLSI architecture for montgomery modular multiplier," ," IEEE Transactions on VLSI system.

[2] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120- 126, 1978.

[3] P. L. Montgomery, "Modular multiplication without trial division," *Math. Comput.*, vol. 44, no. 170, pp. 519–521, Apr. 1985.K. Elissa, "Title of paper if known," unpublished.

[4] Christo Ananth, H.Anusuya Baby, "Encryption and Decryption in Complex Parallelism", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 3, Issue 3, March 2014,pp 790-795

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

[6] 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.