IJMTARC – VOLUME – V – ISSUE – 20 , DEC , 2017 ISSN: 2320-1363

HIGH PERFORMANCE FSI ARCHITECTURE FOR MONTGOMERY MULTIPLICATION UNIT

1.KARISHMA KISHWAR,2.NIHARIKA

1.Pg Scholar, Department of ECE, Vaagdevi College of Engineering, Bollikunta Warangal, Telangana

2.Assistant Professor, Department of ECE, Vaagdevi College of Engineering, Bollikunta Warangal, Telangana

ABSTARCT: In present generation Cryptography plays a crucial role in security purpose. Security comes mostly with three parameters Confidentiality, Integrity and authentication. All these terms are important for a data to be secured. For hardware implementation of this process Montgomery Modular Multiplication is used, for encryption process in public key cryptography. This paper is discussing about the Semi Carry Save based Montgomery Modular Multiplication (SCS-MM2), with high speed performance. In this Paper, we propose a modified SCS based Montgomery modular multiplication (SCS-MM2) with a Reversible Carry Save Adder (RCSA) using peres gates, so that the performance can be increased, and its simulation and synthesis results are presented. Previously, the radix-2 Montgomery modular Multiplication (MM) architecture was implemented for Basic MM, Full Carry Save Montgomery Modular multiplication (FCS-MM) and the Basic SCS-MM1. The proposed Radix-2 modified SCS-MM2 describes high performance architecture and its results are shown for 128bit length. The resultant architecture is simulated using Modelsim, design verification and synthesis results are done using Xilinx ISE. The proposed architecture is compared with the existing SCS-MM2 it can achieve high performance.

I. Introduction

Cryptography is a method of storing and transmitting data in a particular form, so that those whom it is intended only can read and process the data. Cryptography is very essential for security purpose in data transmission. For its hardware implementation Montgomery modular Multiplication algorithms is used. Mostly in Public Key Cryptography this logic is used in data encryption process. Montgomery algorithm can be classified into two types

IJMTARC – VOLUME – V – ISSUE – 20 , DEC , 2017 ISSN: 2320-1363

based on its operation. They are Full-Carry-Save Montgomery modular Multiplication (FCS-MM) and Semi-Carry-Save Montgomery modular multiplication (SCS-MM1) forms. In FCS-MM both the obtained carry and sum are considered as outputs. In SCS-MM only the sum which was obtained is considered as output. When compared to FCS-MM, SCS-MM is having a low area because of less number of adder levels in the basic algorithm. In this paper we discuss about the Modified SCS-MM2 architecture and analyse it for 128-bit inputs. In SCS-MM algorithm it has three input A, B, N, and S as sum output .A is a Multiplicand, B is multiplier and N is modulus. There are some rules for considering the inputs. They are length of the inputs should be same. Modulus value should be always greater than the multiplicand and multiplier.

II. LITERATURE SURVEY

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences: 1. Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, only he knows the corresponding decryption key. 2. A message can be “signed” using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in “electronic mail” and “electronic funds transfer” systems. A message is encrypted by representing it as a number M, raising M to a publicly specified power e, and then taking the remainder when the result is divided by the publicly specified product, n, of two large secret prime numbers p and q. Decryption is similar; only a different, secret, power d is used, where $e \cdot d \equiv 1 \pmod{(p-1)\cdot(q-1)}$. The security of the system rests in part on the difficulty of factoring the published divisor, n. Key Words and Phrases: digital signatures, public-key cryptosystems, privacy, authentication, security, factorization, prime number, electronic mail, messagepassing, electronic funds transfer, cryptography.

III. PROPOSED MONTGOMERY MULTIPLICATION

In this section, we 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 also improved while maintaining the advantages of short critical path delay and low hardware complexity

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, we can precompute D = B + N and reuse the one-level CSA architecture to perform B+N and the format conversion. The Q_L circuit decides the qi value according to step 7 . The carry propagation addition operations of B + N and the format conversion are performed by the one-level CSA architecture of the MSCS-MM multiplier through repeatedly executing the carry-save addition (SS, SC) = SS + SC

IJMTARC – VOLUME – V – ISSUE – 20 , DEC , 2017 ISSN: 2320-1363

+ 0 until SC = 0. In addition, we also precompute Ai and qi in iteration i−1 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. Therefore, the critical path delay of the MSCS-MM multiplier can be reduced into TMUX4 + TFA. However, in addition to performing theA. 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, we can precompute D = B + N and reuse the one-level CSA architecture to perform B+N and the format conversion. Fig. 7(a) and (b) shows the modified SCS-based Montgomery multiplication (MSCS-MM) algorithm and one possible hardware architecture, respectively. The Zero_D circuit in Fig. 7(b) 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 according to step 7 of Fig. 7(a). The carry propagation addition operations of B + N and the format conversion are performed by the one-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 precompute 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. Therefore, the critical path delay of the MSCS-MM multiplier can be reduced into TMUX4 + TFA. However, in addition to performing the

Algorithm Modified SCS-MM:
Modified SCS-based Montgomery multiplication

Inputs : A, B, N (modulus)
Output : $SS[k+2]$

1. $(SS, SC) = (B + N + 0)$;
2. while $(SC != 0)$
3. $\quad (SS, SC) = (SS + SC + 0)$;
4. $D = SS$;
5. $SS[0] = 0$; $SC[0] = 0$;
6. for $i = 0$ to $k + 1$ {
7. $\quad q_i = (SS[i]_0 + SC[i]_0 + A_i \times B_0) \bmod 2$;
8. $\quad$ if $(A_i = 0$ and $q_i = 0)$ $x = 0$;
9. $\quad$ if $(A_i = 0$ and $q_i = 1)$ $x = N$;
10. $\quad$ if $(A_i = 1$ and $q_i = 0)$ $x = B$;
11. $\quad$ if $(A_i = 1$ and $q_i = 1)$ $x = D$;
12. $\quad (SS[i+1], SC[i+1]) = (SS[i] + SC[i] + x) / 2$;
13. }
14. while $(SC[k+2] != 0)$
15. $\quad (SS[k+2], SC[k+2]) = (SS[k+2] + SC[k+2] + 0)$;
16. return $SS[k+2]$;

IV. SCS-MM2 Algorithm

IJMTARC – VOLUME – V – ISSUE – 20 , DEC , 2017 ISSN: 2320-1363

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 ith 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 ];
```

Fig1. 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".

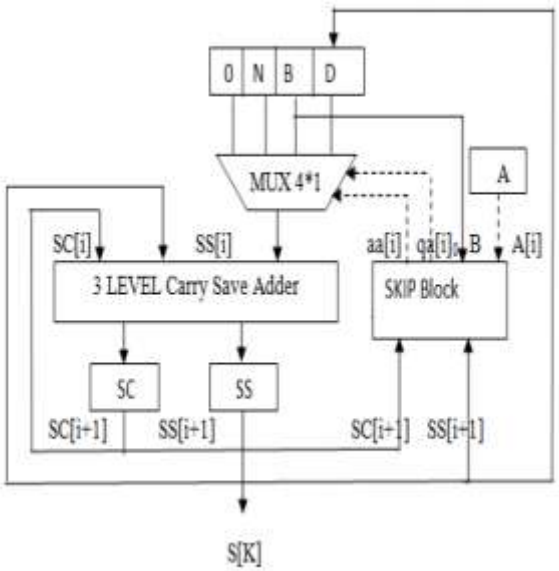

Fig.1(a) Block diagram of SCS-MM2 algorithm.

IJMTARC – VOLUME – V – ISSUE – 20 , DEC , 2017 ISSN: 2320-1363

V. Conclusion

The proposed SCS-MM2 (Semi Carry Save Montgomery modular multiplication) for radix-2 architecture reduces number of critical path delay when compared to the existing logic. The SCS-MM2 (Semi Carry Save Montgomery modular multiplication) architecture is simulated using Modelsim and design verification, area timing report is done using Xilinx ISE 10.1. Finally, the proposed architecture can achieve reduced critical path, and increases the speed of operation.

References

[1]. KUANG et.al."Low Cost High Performance VLSI Architecture for Montgomery MM", IEEE Trans. Very Large Scale Inegr. (VLSI) Syst., vol. 24, issue: 2,pp434-445 , Feb.2016.

[2]. 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.

[3]. Chandra.K and Kumar.P, "Optimization of RSA Processors Using Multiplier", International Journal of Computer Trends and Technology (IJCTT) - Vol-4, Issue-5-May 2013.

[4]. S.Ashwini, P.Thirapaiah "A High-Speed Montgomery Modular Multiplication Algorithm To Reduce The Energy Consumption Based On RSA Cryptosystem", Proc. International Journal of Engineering and Computer Science (IJECS)- Vol-4, Issue-10 Oct 2015,pg-14653-14658.

[5]. Alan Daly and William Marnane "Fast Montgomery Modular Multiplication and RSA Cryptographic Processor Architecture".

[6]. Santharubavagini.K , Abirami.C and Jegadeeshwari,"MBRFA Circuits for High-Throughput and Low Latency Montgomery Modular Multipliers ", International journal of Communication and Computer Technologies (IJCCTS), vol-02,No-09,Issue-02 March 2014

[7]. Jhing-Fa Wang, Po-Chuan Lin and Ping-Kun Chiu "A Staged Carry-Save-Adder Array for Montgomery Modular Multiplication"

[8]. Dr.Deepa Jose , Nathimugil.J and Abida Begum ,"Implementation of Optimized Montgomery modular Multiplier on FPGA", International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST) Vol. 2, Issue 4, April 2016.

IJMTARC – VOLUME – V – ISSUE – 20 , DEC , 2017 ISSN: 2320-1363

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

[10]. David Narh Amanor, Christof Paar, Jan Pelzl and Viktor Bunimov, Manfred Schimmler ,"Efficient Hardware Architectures For Modular Multiplication On Fpgas".

[11]. Harmeet Kaur1, Mrs.Charu Madhu,"Montgomery Multiplication Methods - A Review" International Journal Of Application Or Innovation In Engineering & Management (Ijaiem) Volume 2, Issue 2, February 2013