Dedicated multiplication hardware on FPGA with low Power

Mr. Udaykumar Nagashettappa
Associate professor ECE Dept.,
BKIT,Bhalki

Mrs.Chamundeshwari Goudappa
Assistant professor CSE Dept
BKIT,Bhalki

Abstract :This paper proposes a basic and effective Montgomery augmentation calculation with the end goal that the minimal effort and elite Montgomery measured multiplier possibly actualized. The furthered multiplier gets and yields data with twofold depiction and uses only a solitary level CSA to avoid the convey proliferation at each development activity. This CSA is bestow to perform operand pre-computation and arrangement change from convey spare association to twofold depiction provoking a lower hardware cost and basic way concede costs consummatingone particular duplication. To beat this issue, a configurable CSA (CCSA), which have one full viper or two semi half adders, is exerted to decrease the extra clock cycles and setup change extensively. Additionally, a framework thatperhaps recognize and skirt the pointless convey spare extension activity in the one-level CCSA plan while keeping up the short basic way delay is created. In this manner, the extra clock cycles for operand pre estimation and game plan change possibly covered up and high throughput possibly procured. Test result shows that the furthered Montgomery Modular Multiplication perhaps accomplish higher execution and critical area– time item upgrade when deviate with past plans.

Keyword: CCSA, Montgomergy, Low power, multiplication.

I. Introduction

In several crypto systems, modular multiplication (MM) with expansive whole numbers is the uttmost classical operation. In this manner, various calculations and equipment execution have been introduced to complete the MM[1] all the ample rapidly, and Montgomery's calculation is a standout amongst the uttmost understood MM calculations.

A few variations of FFT strategy have been furthered pointing at the evasion zero-cushioning. Phatak and Goff's strategy plays out the Montgomery measured decrease by one straight convolution and one cyclic convolution. Their strategy could incompletely keep distant from the zero-cushioning issue, since just cyclic convolution conceivably registered without zero-cushioning[3].

It [4] depicts the plan and usage of low power particular multiplier of RSA and equalizations its zone and speed. By enhancing Montgomery particular duplication calculation, advancing basic way and utilizing a few low power techniques, this paper accomplishes low power and additionally fast execution. The outline is actualized utilizing SMIC 0.13um CMOS process, the normal power utilization is 106uW at 13.56MHZ when executing 1024-piece operations, the zone is around 0.17mm2and an opportunity to complete particular increase are 1412 clock cycles, such fantastic property make it appropriate for RSA operation.

It [5] Montgomery multipliers of carry save snake (CSA) design require a full expansion to change over the carry save portrayal of outcome into a regular frame. In this paper, we reuse the CSA engineering to play out the outcome arrange change, which prompts little zone and quick speed. The aftereffects of execution on FPGAs demonstrate akin the new Montgomery multiplier is around 113.4 M_{bit}/s for 1024-piece operands at a clock of 114.2 MHz

Montgomery[6] confined increment is one of real tasks exerted as a piece of cryptographic estimations, for instance, RSA and Elliptic Curve Cryptosystems. At CHES 1999, Tenca and Koc¸ furthered the Multiple-Word Radix-2 Montgomery Multiplication (MWR2MM) figuring and introduced a now-commendable outline for consummating Montgomery duplication in gear. With parameters updated for minimum inaction, this plan plays out a singular Montgomery increment in approximately2nclock cycles, where n is the range of operands in bits. In this paper, we propose two new hardware structures thatperhaps play out a comparable activity in generally n clock

cycles with outmost a comparative clock period. These two plans rely upon pre preparing midway results using two possible doubts concerning the utmost basic bit of past word. These two models defeat the primary building of Tenca and Koc¸ to extent the thing dormancy times locale by 23 and 50 percent, independently, for a couple of utmost ordinary operand sizes exerted as a piece of cryptography.

II. Methodology

Around there, we propose another SCS-based Montgomery MM figuring to lessen the fundamental route deferral of Montgomery multiplier. Likewise, the drawback of ample clock cycles for consummating one duplication is also improved while keeping up the upsides of short fundamental way delay and low gear versatile quality.

A. Critical Path Delay Reduction

The essential path deferral of SCS-based multiplier possibly diminished by joining the advantages of FCS-MM-2 and SCS-MM-2. That is, we perhapspre register D=B+N and reuse the one-level CSA building to perform B+N and the setup change. Fig. 1 exhibits the modified SCS-based Montgomery increment (MSCS-MM) figuring and one possible gear outline, exclusively. The Zero_ D circuit in Fig. 1 is exerted to recognize whether SC is equal to zero, which possibly master using one NOR activity. The Q_L circuit picks the qi regard according to stage 7 of algorithm stated below.The extension tasks of B+N and the setup change are performed by the one-level CSA designing of MSCS-MM multiplier through over and again executing the convey spare development (SS, SC) =SS+SC+0 until SC=0.

In addition, we furtheramplepre process A_i and q_i in cycle i−1 (this will be cleared up ample doubtlessly in Section III-C) so they possibly exerted to rapidly pick the pined for information operand from 0, N, B, and D through the multiplexer M3in accentuation I. Forth these lines, the essential route delay of MSCS-MM multiplier possibly diminished into TMUX4+TFA. Regardless, despite playing out the three-input convey spare additions [i.e., stage 12 of algorithm stated below)] k +2 times, various extra clock cycles are endorsed to perform B+N and the arrangement change through the one-level CSA designing since they ought to be performed once in every MM.

Algorithm for MM

Inputs: A,B,N(modulus)

Output: SS[k+2]

1. (SS,SC) =(B+N+0);
2. While(SC!= 0)
3. (SS,SC)=(SS+SC+0);
4. D=SS;
5. SS[0]=0;. SC[0]=0;
6. for i=0 to k-1 {
7. qi=(SS[i]o + SC[i]o + Ai*Bo)mod2;
8. if(Ai=0 and qi=0) x=0;
9. if(Ai=0 and qi=1) x=N;
10. if(Ai=1 and qi=0) x=B;
11. if(Ai=1 and qi=1) x=D;
12. (SS[i+1],SC[i+1])=(SS[i]+SC[i]+x)/2;
13. }
14. While (SC[k+2]!=0)
15. (SS[k+2],SC[k+2])=
16. (SS[k+2]+ SC[k+2]+0);

return SS[k+2];

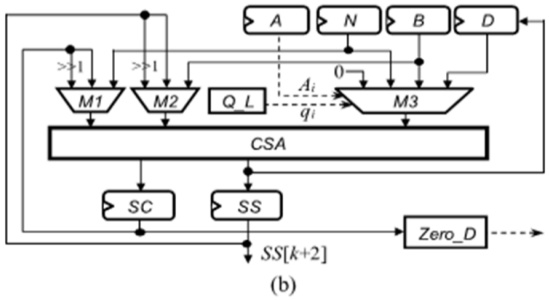

Fig 1. MSCS- MM Multiplier.

B. Clock Cycle Number Reduction

To decrease the clock cycle number, a CCSA outline which perhaps perform one three-input convey spare development or two serial two-input convey spare increases isfurthered to substitute for one-level CSA designing in Fig.1. Fig. 2(a) shows two cells of one-level CSA outline in Fig. 1, each cell is one general FA which

perhaps play out the three-input convey spare development. Fig. 2(b) exhibits two cells of furthered configurable FA (CFA) circuit. In case α =1, CFA is one FA and perhaps perform one three-input convey spare development (implied as 1F_CSA). Else, it is two half-adders (HAs) and perhaps perform two serial two-input convey spare builds (meant as 2H_CSA), as showed up in Fig. 2(c). For this circumstance, G1 of $CFA_j$and$G2ofCFAj+1$in Fig. 2(b) will go about as HA1 j in Fig. 2(c), and G3, G4, and G5 of CFA_jin Fig. 2(b) will bear on as HA2j in Fig. 2(c). Furtherample, we modify the 4-to-1 multiplexer M3in Fig. 1 into an unraveled multiplier SM3as showed up in Fig. 2(d) in light of reality that one of its wellsprings of data is zero, where ~ means the INVERT activity. Note that M3has been supplanted bySM3in the furthered one-level CCSA designing showed up in Fig. 2(b) According to concede extent showed up in Table II,TSM3 (i.e., 0.68×TFA)is vague to TMUX3(i.e., 0.63×TFA) and TMUXI2 (i.e., 0.23 ×TFA) is ample diminutive than TXOR2 (i.e., 0.34×TFA). Thusly, the essential path deferment of furthered one-level CCSA outline in Fig. 2(b) is estimated to that of one-level CSA designing in Fig. 2(a).

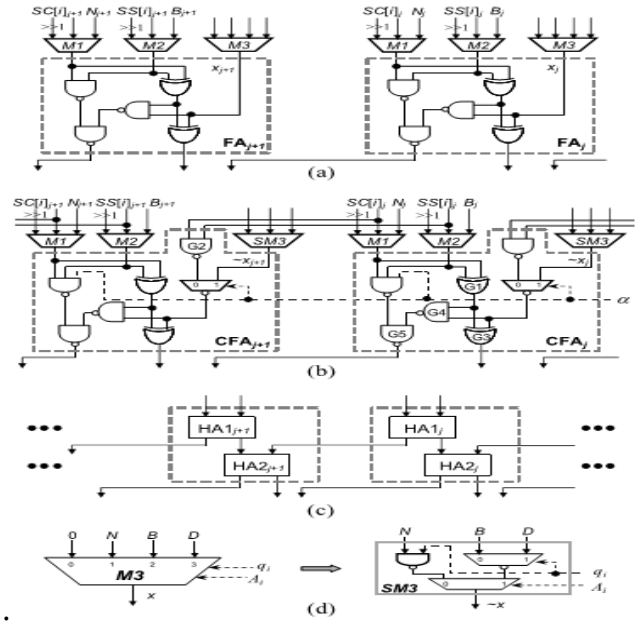

Fig.2. (a) Conventional FA circuit (b) Furthered CFA circuit. (c) Two serial HAs. (d) Simplified multiplexer SM3

Therefore, the computation of (1) in cycle I possibly skipped if we direct right move the yields of one-level CSA outline in the $(I-1)_{th}$ accentuation by useful in vain positions (i.e., discognate by 4) instead of one piece position (i.e., isolated by 2) when Ai = q_i = 0and $SS[i]_0$=$SC[i]_0$=0. Likewise, the banner skipi+1 exerted as a piece of thei_{th} cycle to exhibit whether the convey spare extension in the (I +1) accentuation will be skipped possibly imparted as

$$skip_{i+1} = \sim(A_{i+1} \vee q_{i+1} \vee SS[i+1]_0)$$

Where, ∨ speaks to OR task. If $skip_{i+1}$ created in i_{th}cycle is 0, the convey spare extension of $(I+1)_{th}$ accentuation won't be skipped. For this circumstance, q_{i+1} and A_{i+1} conveyed in ithaccentuation possibly set distant in FFs and after thatbestow to speedy pick the estimation of x_{in} the $(i+1)_{th}$cycle.

C. Quotient Pre computation

As said above A_{i+1}, A_{i+2}, q_{i+1},andq_{i+2} must be known in ithemphasis for avoiding the superfluous operation in the $(i+1)_{th}$ cycle. It is anything but difficult to obtain$A_{i+1}$and A_{i+2}in ithcycle. The remainder q_{i+1} conceivably figured in ithemphasis like stride 7 of Fig. 3.1(a) as takes after

$$q_{i+1} = (SS[i+1]_0 + SC[i+1]_0 + A_{i+1} \times B_0) \bmod 2. \quad (3)$$

Be that as it may, $SS_{[i+1]}0$and$SC_{[i+1]}0$are inaccessible until (1) is finished, as appeared in Fig. 3. Subsequently, the basic way of Montgomery multiplier in Fig. 3.1(b) will be generally stretched if (3) is straightforwardly employed to create q_{i+1}in ithcycle. To keep distant from this situation ,N, B, and Dare adjusted as takes after with the goal that $SS_{[i+1]}0$, $SC_{[i+1]}0$, q_{i+1},andq_{i+2} conceivably immediately originatedin ithcycle. Since modulus Nis an odd number and is comprehended the ith cycle just when q_i is equivalent to one, it is discovered that no less than a proliferated convey 1 is originated since N0 is equivalent to one. Forth these lines, we perhaps straightforwardly utilize the incentive as appeared in (4) rather than N to fulfill the procedure of Montgomery MM. A short time later: 0 must be equivalent to zero.

$$\hat{N} = \begin{cases} N+1, & \text{if } N_{1:0} = 11 \\ 3N+1, & \text{if } N_{1:0} = 01. \end{cases} \quad (4)$$

In addition, we employ^ B=8 Binstead of B to guarantee that 2:0is equivalent to zero so A_{i+1}×B_0in (3) conceivably disposed of and the calculation ofq_{i+2}perhaps likewise be disentangled. Note that

three additional time cycles toward the finish of MM for figuring division by two are important to keep up the rightness of Montgomery MM since Bis supplanted with 8B.If N and B are supplanted by and the created 1:0(= +)must be equivalent to zero. After N, B, and D are supplanted andwe perhaps guarantee that the two LSBs of variable x(i.e., x1:0)in (1) must be equivalent to zero. Accordingly, the convey esteem SC[i+1]0in Fig. 3 is equivalent to $SS_{[i]}0 \wedge SC_{[i]}0$since$x_0=0$

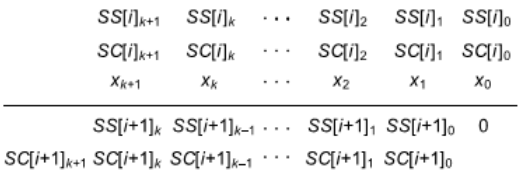

Figure. 3. Three-to-two carry-save addition at the i_{th}iteration of Fig. 1.

D. Furthered Algorithm and Hardware Architecture

On the bases of basic way postpone lessening, clock cycle number decrease, and remainder precomputation specified over, another SCS-based Montgomery MM calculation (i.e., SCS-MM-New calculation appeared in Fig. 4) utilizing one-level CCSA engineering is furthered to fundamentally decrease the prescribed clock cycles for finishing one MM. As appeared in SCS-MM-New calculation, steps 1– 5 for creating and are first performed. Note that in light of fact that q_{i+1} and q_{i+2} must be created in ithemphasis, the iterative file I of Montgomery MM will begin from −1 rather than 0 and the relating starting estimations of and must be set to 0.

NEW FURTHERED SCS-BASED MONTGOMERY MM ALGORITHM:

Inputs :A,B,N(modulus)

Output:SS[k+5]

1. ^B=B<<3;^A=0;^q=0;skipi+1=0;
2. (SS,SC) =(if _CSA ^B+^N+0);
3. While(SC!=0)
4. (SS,SC)=2H_CSA(SS,SC);
5. ^D=SS;
6. i=-1,SS[-1]=0;. SC[-1]=0;
7. while (i<=k+4) {
8. if(^Ai=0 and ^qi=0) x=0;
9. if(^Ai=0 and ^qi=1) x=^N;
10. if(^Ai=1 and ^qi=0) x=^B;
11. if(^Ai=1 and ^qi=1) x=^D;
12. (SS[i+1],SC[i+1])=IF_CSA(SS[i]+SC[i]+x)>>1;
13. compute qi+1,qi+2,skipi+1 by (5),(7) and(8);
14. If(skipi+1 = 1){
15. SS[i+2]=SS[i+1])>>1;(SC[i+2]=SC[i+1])>>1;
16. ^q=qi+2;^A=Ai+2,i=i+2;
17. {
18. else {
19. ^q=qi+1;^A=Ai+1,i=i+1;
20. }
21. }
22. ^q=0;^A=0;

23. While (SC[k+5]!=0)

24. (SS[k+5]=SC[k+5])=2H_(SS[k+5]+ SC[k+5]);

25. return SS[k+5];

Fig. 4(a). SCS-MM-New algorithm

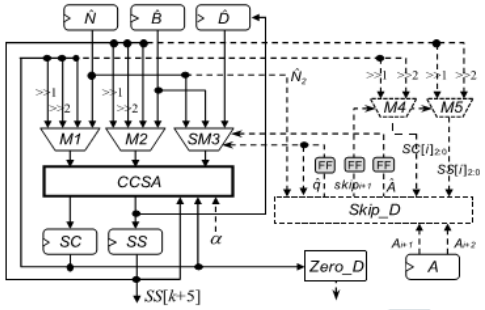

Fig.4 (b).SCS-MM-New multiplier.

Toward the start of Montgomery duplication, the FFs put distant $skip_{i+1}$, , are first reset to 0 as appeared in step 1 of SCS-MM-New calculation with the goal that = + conceivably registered by means of one-level CCSA design. When playing out the while circle, the skip detectorSkip_DshowninFig.3.6is employed to create $skip_{i+1}$,and .The Skip_D is made out of four XOR gates, three AND gates, one NOR gate, and two 2-to-1 multiplexers. It initially creates the q_{i+1}, $q_{i+2}$and $skip_{i+1}$signal in thei_{th} emphasis as indicated by (5), (7), and (8), individually, and after that chooses the right ˆ qand as per $skip_{i+1}$. Toward the finish of the ith emphasis and $skip_{i+1}$must be put distant to FFs.

III. Experimental results

The schematic flow of the proposed system is as shown below fig 5.

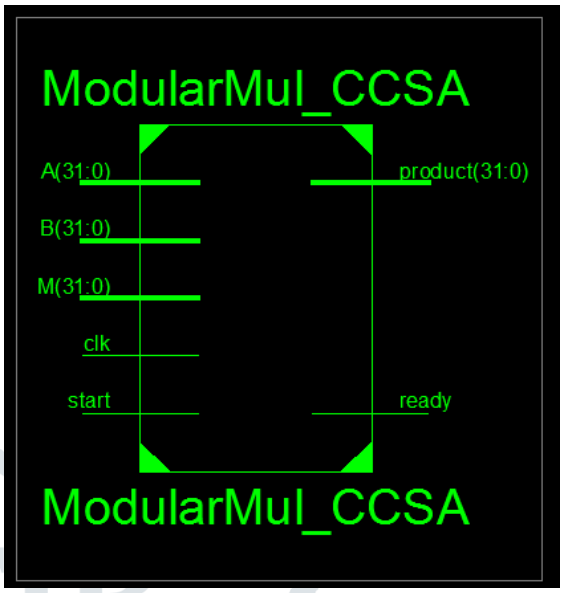

Fig 5. Schematic view of proposed system.

The RTL schematic view is as shown in below fig 6.

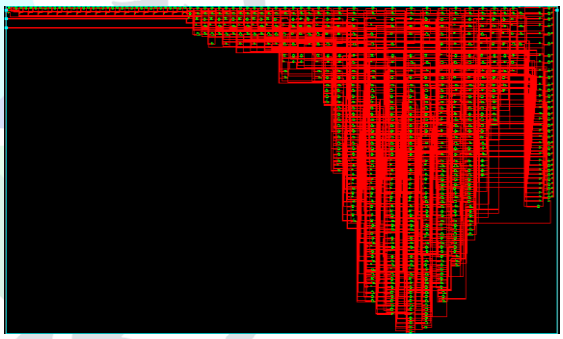

Fig 6. RTL view schematic

The simulation shows the multiplication of two operands of 32 bit in length. The simulation results shows that the power consumed by the proposed system is less.

The results are shown in Fig 7.

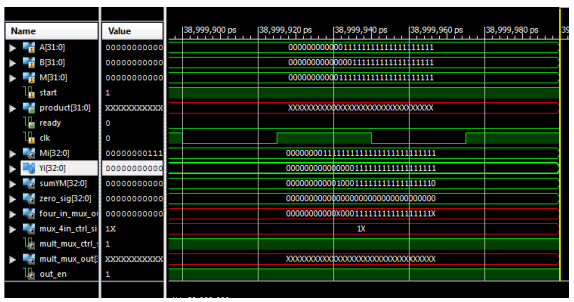

Fig 7. Simulation of multiplication of two operands.

The device utilization by the device is as shown in the Fig 8. The fig 9 shows the total delay generated from the device which is 10.08 nano seconds

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	286	4656	6%
Number of Slice Flip Flops	75	9312	0%
Number of 4 input LUTs	539	9312	5%
Number of bonded IOBs	131	190	68%
Number of GCLKs	1	24	4%

Fig 8. Device utilization by hardware

```
  MUXCY:CI->O          32   0.399   1.225  Mcompar_product_sig_cmp_lt0000_
  LUT3:I0->O            1   0.612   0.357  product_sig<9>1 (product_9_OBUF
  OBUF:I->O                 3.169          product_9_OBUF (product<9>)
----------------------------------------
Total                      10.088ns (7.899ns logic, 2.189ns route)
                                    (78.3% logic, 21.7% route)
```

Fig 9. Delay generated by the proposed system.

IV. Conclusion

FCS-based multipliers keep up the data and yield operands of Montgomery MM in the convey spare design to escape from the association change, provoking less clock cycles yet greater range than SCS-based multiplier. To redesign the execution of Montgomery MM while keeping up the low gear diserse quality, this venture has changed the SCS-based Montgomery increment figuring and furthered a negligible exertion and predominant Montgomery specific multiplier. The furthered multiplier exerted one-level CCSA outline and maintained a strategic distance from the trivial convey spare development tasks, all things studious, diminish the essential way delay and recommended clock cycles for consummating one MM activity. Test comes to fruition showed akin the furthered approaches are point of fact prepared for enhancing the execution of radix-2 CSA-based Montgomery multiplier while keeping up low hardware multifaceted nature.

REFERENCES

[1] N. Koblitz, "Elliptic curve cryptosystems," Math. Comput., vol. 48, no. 177, pp. 203–209, 1987.

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

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

[4] H. Zhengbing, R. M. Al Shboul, and V. P. Shirochin, "An efficient architecture of 1024-bits c for RSA cryptosystem based on modified Montgomery's algorithm," in Proc. 4th IEEE Int. Workshop Intell. Data Acquisition Adv. Comput. Syst., Sep. 2007, pp. 643–646.

[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] S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, "Energy-efficient high-throughput Mmontgomerymodular multipliers for RSA cryptosystems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 11, pp. 1999–2009, Nov. 2013.

[7] BAHARUDDIN BIN ISMAIL, "DESIGN AND DEVELOPMENT OF UNIPOLAR SPWM SWITCHING PULSES FOR SINGLE PHASE FULL BRIDGE INVERTER APPLICATION", Thesis submitted in fulfilment of the requirements for the degree of Master of Science, May 2008.

[8] H.W. van der Broeck, H.-C. Skudelny, and G.V. Stanke, "Analysis and realization of a pulsewidth modulator based on voltage space vectors," IEEE Transactions on Industry Applications, vol.24, pp. 142-150, 1988.

[9] Richard E. Haskell, Darrin M. Hanna, "Introduction to Digital Design Using Digilent FPGA Boards", Published by LBE Books, LLC, 2009.