International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Enhanced Vlsi Architecture For Montgomery Modular Multiplication In Digital Filters

*E.Shirisha ** D.SrinivasRao ***Mr.G.Babu



*M.Tech Dept of E.C.E, Vaagdevi College Of Engineering
**Assistant. Prof Dept of E.C.E, Vaagdevi College of Engineering
**Associate. Prof Dept of E.C.E, Vaagdevi College of Engineering

Abstract:

The multiplier receives and outputs the data with binary representation and uses only one-level Carry Save Adder (CSA) to avoid the carry propagation at each addition operation. A famous approach to implement modular multiplication in hardware circuits is based on the Montgomery modular multiplication algorithm since it has many advantages. To speed up the encryption/decryption process, many high-speed Montgomery modular multiplication algorithms and hardware architectures employ carry-save addition.This CSA is also used to perform operand pre computation and format conversion from the carry save format to the binary representation, leading to a low hardware cost and short critical path delay at the expense of extra clock cycles for completing one modular multiplication. To overcome the weakness, a Configurable CSA (CCSA), which could be one full-adder or two serial half-adders, is proposed to reduce the extra clock cycles for operand pre computation and format conversion by half. The mechanism that can detect and skip the unnecessary carry-save addition operations in the one-level CCSA architecture while maintaining the

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

short critical path delay is developed. The extra clock cycles for operand pre computation and format conversion can be hidden and high throughput can be obtained

Keywords--Carry-save addition, low-cost architecture, Montgomery modular multiplier, public-key cryptosystem.

INTRODUCTION

In Many public-key cryptosystems, modular multiplication (MM) with large integers is the most critical and time-consuming operation.Given two integers *a* and*b*, the classical modular multiplication algorithm computes *ab* mod *N*. Montgomery multiplication works by transforming *a* and *b* into a special representation known as Montgomery form. For a modulus *N*, the Montgomery form of *a* is defined to be *aR* mod *N* for some constant *R* depending only on *N* and the underlying computer architecture. If *aR* mod *N* and *bR* mod *N* are the Montgomery forms of *a* and *b*, then their Montgomery product is *ab R* mod *N*. Montgomery multiplication is a fast algorithm to compute the Montgomery product. Transforming the result out of Montgomery form yields the classical modular product *ab* mod *N*.

Therefore, numerous algorithms and hardware implementation have been presented to carry out the MM more quickly, and Montgomery's algorithm is one of the most well-known MM algorithms. Montgomery's algorithm determines the quotient only depending on the least significant digit of operands and replaces the complicated division in conventional MM with a series of shifting modular additions to produce $S = A \times B \times R^{-1} \pmod{N}$, where N is the k-bit modulus, R^{-1} is the inverse of R modulo N, and $R = 2^k \bmod N$. However, the three-operand addition in the iteration loop of Montgomery's requires long carry propagation for large operands in binary representation. To solve this problem, several approachesof based on carry-save addition were proposed to achieve a significant speedup of Montgomery MM. Based on the representation of input and output operands, these approaches can be roughly divided into semi-carry-save (SCS) strategy and full carry-save (FCS) strategy. In the SCS strategy the input and output operands (i.e., *A*, *B*, *N*, and *S)*

of the Montgomery MM are represented in binary, but intermediate results of shifting modular additions are kept in the carry-save format to avoid the carry propagation. However, the format conversion from the carry-save format of the final modular product into its binary representation is needed at the end of each MM. This conversion can be accomplished by an extra carry propagation adder (CPA) or reusing the carry-save adder (CSA) architecture iteratively. Contrary to the SCS strategy, the FCS strategy maintains the input and output operands *A*, *B*, and *S* in the carry-save format, denoted as (*AS*, *AC*), (*BS*, *BC*), and (*SS*, *SC*), respectively, to avoid the format conversion, leading to fewer clock cycles for completing a MM. Nevertheless, this strategy implies that the number of operands will increase and that more CSAs and registers for dealing with these operands are required. Therefore, the FCS-based Montgomery modular multipliers possibly have higher hardware complexity and longer critical path than the SCS-based multipliers.

2.EXISTING ARCHITUCTURES

2.1.CSA BASED MONTGOMERY MULTIPLICATION

Montgomery multiplication algorithm is the most efficient algorithm available. The main advantage of Montgomery algorithm is that it replaces the division operation with shift operations. During two decades many alternativforms of Montgomery algorithms are introduced. These architectures use carry save addition. The work n [5] and [13]presented two types of Montgomery algorithms whichuse Carry Save Adder (CSA). One of the two types used four-totwo CSA and the other used five-to-CSA. They had given a brief comparison between these two versions of Montgomerymultipliers. They had found that the multiplier using four-to-two CSA architecture has shorter critical path than that offive-to-two CSA multiplier. But extra storage elements and multiplexers are required for 4-to-2 architecture whichprobably increases the energy consumption. The work in [6] proposed a Montgomery multiplication algorithm usingpipelined carry save addition to shorten the critical path delay of five-to-two CSA. This method also required additional pipeline registers and multiplexers which will increase the area. Ming Der Shieh presented [7] a new algorithm for high speed modular

multiplication. This new Montgomerymultiplier performs modular reduction in a pipelined fashion, so that the critical path delay is reduced from the four-to-twoto three-to-two carry-save addition. Then it requires additional pipeline registers to store intermediate values. All the aboveworks were not discussed about the energy consumption. Several previous works [8], [9] have developed techniques to reduce the power/energy consumption ofMontgomery multipliers. In [8], some latches named glitch blockers are located at the outputs of somecircuit modules toreduce the spurious transitions and the expected switching activities of high fan-out signals in the radix-4 scalableMontgomery multiplier. ShiannRongKuang [9] triedto reduce the energy consumption of CSAs and registers in theCSA-based Montgomery multipliers via a new technique. They modified the CSA based Montgomery multiplier algorithmand as a result of this, the number of clock cyclesrequired to complete the multiplication is largelydecreased. To achievefurther energy reduction, they have adjusted the internal structure of barrel register full adder andthen applied the gated clock design technique. But this energy efficient algorithm increased the total area of the design.

In order to reduce the area, we are presenting a new algorithm which can modify the CSA based algorithm in [9].Before introducing the area efficient algorithm, let us see the mathematics behind the Montgomery modular multiplication

MONTGOMERY MODULAR MULTIPLICATION

Modular multiplication of two integers X and Y, simply performs,

$$Z = X.Y \bmod M \quad (1)$$

Here X,Y and M are n- bit numbers and M should be greater than X and Y. Instead of computing X. Y, the Montgomery multiplication [10] algorithm computes,

$$Z' = MP(X, Y, M) = X.Y.2^{-n} \bmod M \quad (2)$$

MP denotes Montgomery Product and sometimes the equation (2) can also be represented as shown below.

$$Z' = MP(X, Y, M) = X.Y.R^{-1} \bmod M \quad (3)$$

Here R = 2n and R-1 is the multiplicative inverse of R mod M. That is,

$$R.R^{-1} \bmod M = 1 \quad (4)$$

In order to perform Montgomery multiplication the numbers should be converted to the Montgomery domain. The conversion to the Montgomery domain is very simple. The conversion from integer domain to Montgomery domain is shown below.

$$X' = X.\ R \bmod M \quad (5)$$

$$Y' = Y.\ R \bmod M \quad (6)$$

Here X' and Y' are the Montgomery forms of X and Y. after completing the multiplication in Montgomery domain, we have to convert it back to the integer domain to obtain the final result. The conversion from the Montgomery product Z' to Z is shown below:

$$Z = MP(Z', 1, M) = Z' .1.\ R\text{-}1 \bmod M \quad (7)$$

To avoid the drawbacks in CSA algorithm carry save is divided into two categories that are:1.SCS- based Montgomery Multiplication

2. FCS-based Montgomery Multiplication

2.2. SCS-Based Montgomery Multiplication

In the SCS strategy the input and output operandsof the Montgomery MM are represented in binary, but intermediate results of shifting modular additions are kept in the carry-save format to avoid the carry propagation. However, the format conversion from the carry-save format of the final modular product into its binary representation is needed at the end of each MM. This conversion can be accomplished by an extra carry propagation adder (CPA) or reusing the carry-save adder (CSA) architecture iteratively.

2.3. FCS-Based Montgomery Multiplication

To avoid the format conversion, FCS-based Montgomery multiplication maintains *A*, *B*, and *S* in the carry save representations (*AS*, *AC*), (*BS*, *BC*), and (*SS*, *SC*), respectively. McIvor *et al.* proposed two FCS based Montgomery multipliers, denoted as FCS-MM-1 and FCS-MM-2 multipliers, composed of one five-totwo(three-level) and one four-to-two (two-level) CSA architecture, respectively. The barrel register full adder (BRFA) consists of two shift registers for storing *AS* and *AC*, a full adder (FA), and a flip-flop (FF). For moredetails about BRFA, On the other hand, the FCS-MM-2 multiplier proposed adds up *BS*, *BC*, and *N* into *DS* and *DC* at the beginning of each MM.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Therefore, the depth of the CSA tree can be reduced from three to two levels. Nevertheless, the FCS-MM-2 multiplier needs two extra 4-to-1 multiplexers addressed by *Ai* and *qi* and two more registers to store *DS* and *DC* to reduce one level of CSA tree. Therefore, the critical path of the FCS-MM-2 multiplier may be slightly reduced with a significant increase in hardware area when compared with the FCS-MM-1 multiplier. Generally speaking, SCS-based multipliers have lower area complexity than FCS-based Montgomery multipliers. However, extra clock cycles for format conversion possibly lower the performance of SCS-based multipliers. To further enhance the performance of the SCS-based multiplier, both the critical path delay and clock cycles for completing one multiplication must be reduced while maintaining the low hardware complexity.

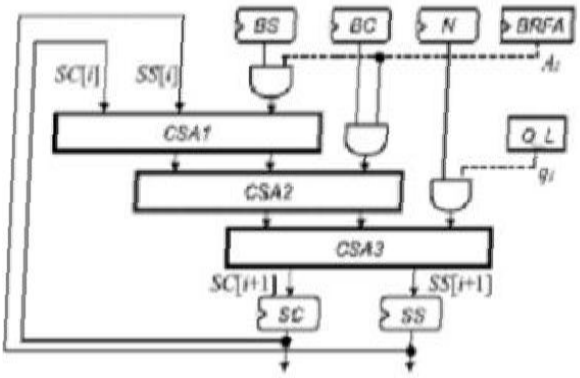


Fig:FCS-MM Block diagram

DRAW BACKS:

1. More Hard ware complexity

2. large area

3. More power consumption

4.High cost

3. Proposed algorithm and hardware architecture

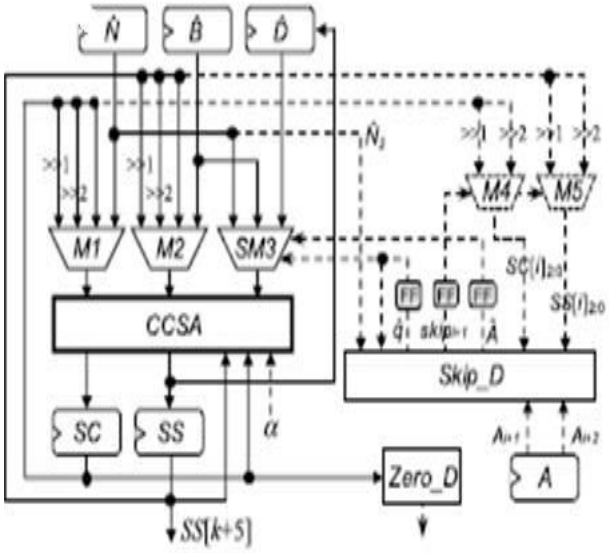
semi carry save multiplier is first used to pre-compute the four-to-two carry save additions. Then the required multiplication can be performed. The modulus N and inputs will be allowed inside the two multiplexers. This partial product is then allowed inside the multiplier.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Those partial outputs then enter into configurable carry save adder, where the carry save addition operation is performed. They are stored in the flip flops temporarily. When another partial output is executed, then that will be stored in the flip flop.

The Skip detector will skip the previous multiplication which is not required in the operation so as to reduce the number of clock cycles. The partial product from SM3 is allowed to the multiplexers M4 and M5. Later on it allows inside the flip flops for temporary storage, then to the skip detector.

The output can be obtained from semi carry. This process is repeated until the output is obtained. The zero detectors can also be used to detect zero in many situations, which is most required. The complexity is very less compared to the previous one.



SCS-MM-New multiplier.

3.1. 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 pre compute $D = B + N$ and reuse the one-level CSA architecture to perform $B+N$ and the format conversion. Figure.1 shows the modified SCS-based Montgomery multiplication (MSCS-MM) algorithm and possible hardware architecture, respectively

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

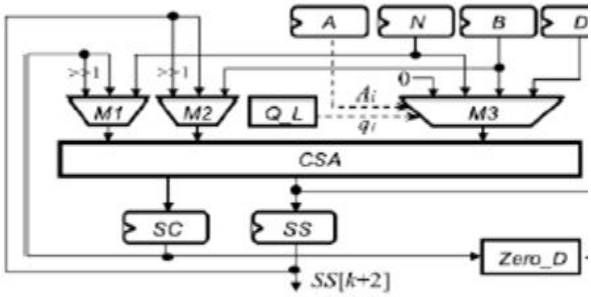


Diagram of Montgomery Modular Multiplie

The Zero_D circuit 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. 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 pre compute *Ai* and *qi* in iteration *i*−1 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.

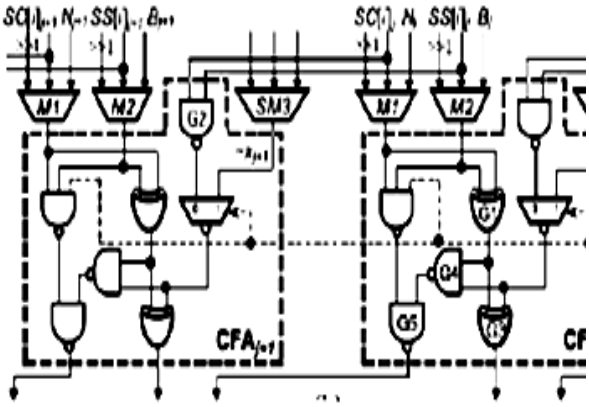
Therefore, the critical path delay of the MSCS-MM multiplier can be reduced into *T*MUX4 + *T*FA. However, in addition to performing the three-input carry-save additions *k* + 2 times, many extra clock cycles are required to perform *B* + *N* and the format conversion via the one-level CSA architecture because they must be performed once in every MM. Furthermore, the extra clock cycles for performing *B*+*N* and the format conversion through repeatedly executing the carry-save addition *(SS, SC)* = *SS*+*SC*+0 are dependent on the longest carry propagation chain in *SS* + *SC*. If *SS* = 111…1112 and *SC* = 000…0012, the one-level CSA architecture needs *k* clock cycles to complete *SS* + *SC*. That is, 3*k* clock cycles in the worst case are required for completing one MM. Thus, it is critical to reduce the required clock cycles of the MSCS-MM multiplier

3.2. Clock Cycle Number Reduction

To decrease the clock cycle number, a CCSA architecture which can perform one three-input carry-save addition or two serial two-input carry-save additions is proposed to substitute for the one-level CSA architecture [4]. Two cells of the one-level CSA architecture in Figure.2each cell is one conventional FA which can perform the three-input carry-save addition. Two cells of the proposed

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

configurable FA (CFA) circuit. If α = 1, CFA is one FA and can perform one three-input carry-save addition (denoted as 1F_CSA).

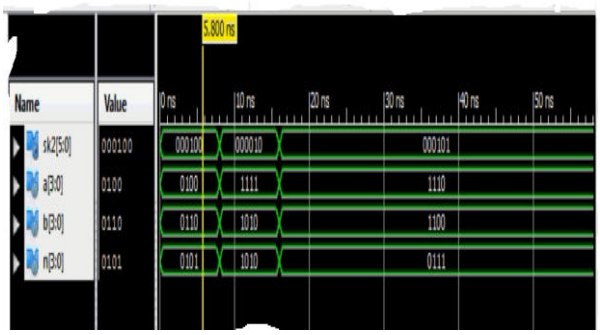


Carry Full Adder Circuit

Otherwise, it is two half-adders (HAs) and can perform two serial two-input carry-save additions (denoted as 2H_CSA). In this case, G1 of CF Aj and G2 of CFAj+1 will act as HA1 j and G3, G4, and G5 of CF Aj will behave as HA2j. Moreover, we modify the 4-to-1 multiplexer M3 into a simplified multiplier SM3 because one of itsinputs is zero, where the INVERT operation. Note that M3 has been replaced by SM3 in the proposed one-level CCSA architecture.

Experimental Results

This is the output of the proposed SCS-MM



Synthesis report of SCS-MM is below:

Logic utilization	**used**	**Available**	**utilization**
No of 4 input LUTs	58	1920	3%
No of occupied slices	30	960	3%
No of logic contains only related logic	30	30	100%
Total No of 4 input LUTs	59	1920	3%

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

FIR Filters:

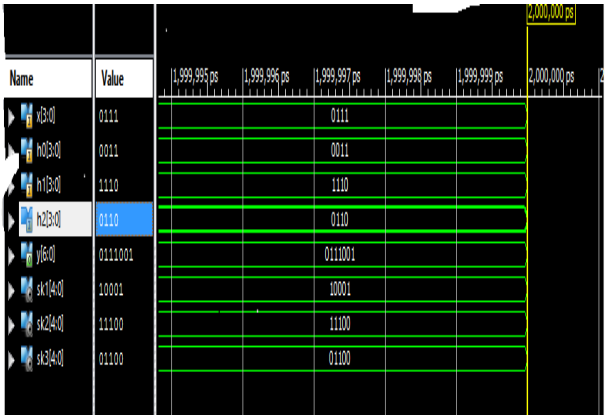
Digital filters arerapidly replacing classicanalog filters.ProgrammableDSP with MACcanbe used to implement digital filters.For high-bandwidth signal processing applications,FPGA technologycan providemultipleMACstoachievethedesired throughput.An FIR withconstant coefficientis anLinearTime-Invariant (LTI) filter. Theoutputof an FIR of order (or length) L, to an input time-series x[n], is given by a finite version of theconvolution sum:

$$y(n) = \sum_{k=0}^{M-1} h(k)\, x(n-k)$$

since h(n) and x(n) are finite duration sequences, their convolution is also finite in duration. The duration of the sequence y(n) is L+M-1.

The modified booth recorder written in Verilog, compiled and simulation using Modelsim.The circuit simulated and synthesized.here Montgomery multiplication is used in filters.when we use filters in crypto systems,then we use Montgomery multiplication in filters to speed up the multiplication.

Here is the output of filter when Montgomery multiplication is used in that



CONCLUSION

To enhance the performance of Montgomery MM while maintaining the low hardware complexity, this paper has modified the SCS-based Montgomery multiplication algorithm a low-cost and high-performance Montgomery modular multiplier. The multiplier used one-level CCSA architecture and skipped the unnecessary carry-save addition operations to largely reduce the critical path delay and required clock cycles for completing one MM operation. FCS-based multipliers maintain the input and output operands of the Montgomery MM in the carry-save

format to escape from the format conversion, leading to fewer clock cycles but larger area than SCS-based Multiplier. In Future, for cryptographers, a cryptographic "break" is anything faster than a brute force performing one trial decryption for each key (see Cryptanalysis). This includes results that are infeasible with current technology. The largest successful publicly known brute force attack against any block-cipher encryption was against a 64-bit RC5 key.

REFERENCES

1. Amber.P, Pinckney.N, and Harris, D. M. "Parallel high-radix Montgomery multipliers,"(2008) in Proc. 42nd Asilomar Conf. Signals, Syst., Comput., pp. 772–776.

2. Bunimov.V, Schimmler.M, and Tolg.B, "A complexity-effective version of Montgomery's algorihm," (2002) in Proc. Workshop Complex.Effective Designs.

3. Gang.F, "Design of modular multiplier based on improved Montgomery algorithm and systolic array," (2006) in Proc. 1st Int. Multi-Symp. Comput.Co mput.Sci., vol. 2. Jun. 2006, pp. 356–359.

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

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

6. McIvor.C, McLoone.M, and McCanny, J. V. "Modified Montgomery modular multiplication and RSA exponentiation techniques,"(2004) IEE Proc.-Comput. Digit.Techn., vol. 151, no. 6, pp. 402–408,.

7. Miyamoto A., Homma N., Aoki, T. and Satoh.A, "Systematic design of RSA processors based on high-radix Montgomery multipliers,"(2011) IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 7, pp. 1136–1146.

8. Neto, J. C. Tenca A. F., and Ruggiero W. V., "A parallel k-partition method to

perform Montgomery multiplication,"(2011) in Proc. IEEE Int. Conf. Appl.-Specific Syst., Archit., Processors, , pp. 251–254.

AUTHOR 1:-

E.SHIRISHA completed her B-tech in S.R ENGINEERING COLLEGE. in 2014 and completed M-Tech in VAAGDEVI COLLEGE OF ENGINEERING.

AUTHOR 2:-

D.SRINIVAS RAO is working as Assistant. professor in Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING.

AUTHOR 3:-

Mr.G.BABU is working as Associate. professor in Dept of ECE, VAAGDEVI COLLEGE OF ENGINEERING.