An UGC-CARE Approved Group-I Journal www.ijearst.co.in

ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

UGC Care Group-I
Vol.03, Issue.01
November-2022
Pages: -1140-1151

DESIGN OF HIGH-SPEED MAC UNIT USING REVERSIBLE GATES FOR DSP APPLICATIONS

[1]GUDALA PAVANI DEVI, [2]K. VIJAYA PRASAD
[1]M. Tech, Dept. of E.C.E, DJR Institute of Engineering & Technology, Gudavalli, Vijayawada, A.P
[2]Professor & H.O.D, Dept. of E.C.E, DJR Institute of Engineering & Technology, Gudavalli, Vijayawada, A.P

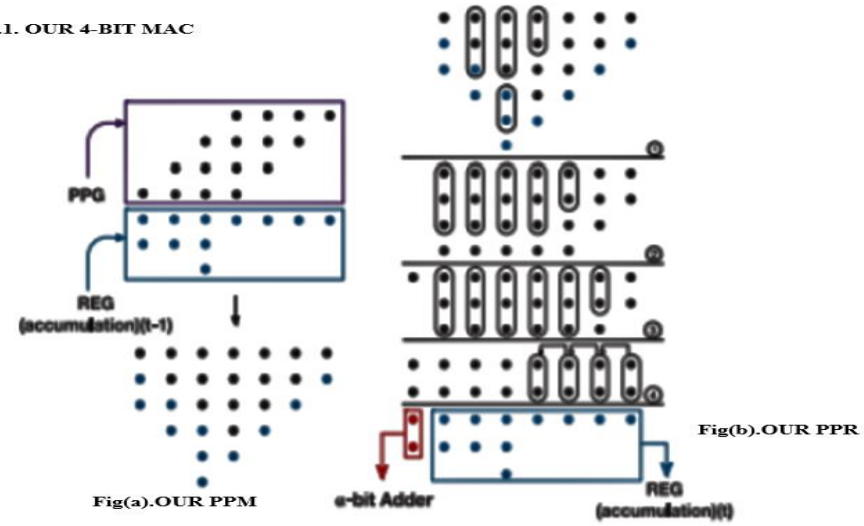
Abstract: Low-power high-speed pipeline MAC architecture is proposed. Carry propagations of additions (including additions in multiplications and additions in accumulations) often contribute to excessive power consumption and large route latency in a typical MAC. Part of the modifications are included into the PPR process in order to address this issue. It is not until the PPR phase of the following multiplication that pieces of greater importance are added and amassed in the suggested MAC architecture. Excess in the PPR process is handled with a small-size adder that keeps track of all carries. Experiments indicate that the suggested MAC design significantly reduces both energy usage and circuit size under the same time restriction compared to earlier efforts.

Keywords: MAC(Multiply accumulate unit), adder, multiplier ETC.,

Introduction: Digital signal processing (DSP) relies heavily on the MAC unit, which is a vital component. The creation of real-time edge applications has grown more popular in recent years. Because of this, the market for high-speed, low-power MAC units is expected to grow. A multiplier and an accumulator are two separate blocks in a typical MAC unit (i.e., an accumulate adder). To prevent overflow, an N-bit multiplier and an accumulator (2N+-1)-bit accumulator (adder) are included in an N-bit MAC unit (caused by long sequences of multiply-accumulate operations). The optimization of the multiplier and also the optimization of the adder have been the subject of many earlier studies. For a multiplier, there are generally three distinct stages. The partial product generation (PPG) method is the initial phase. In the case of an unsigned multiplication, AND gates may be employed to construct a 1-bit matrix (PPM). Partially reducing the product (PPR) is the next phase. The PPM may be reduced to two rows by utilising the Dadda tree strategy or the Wallace tree approach. The last step is the adding of the third step. The last two rows are added together using an adder (referred to as the final adder). The final addition of an N-bit multiplier requires a (2N-1)-bit adder. Tradeoffs between latency, space and power may be made using various adder topologies. The multiplier and the accumulator (adder) may also be replaced with other designs in order to create a variety of MAC unit types. There are comparisons between several MAC unit types in terms of latency, area, and power in.

A substantial amount of power is used, and the route latency increases, when additions are carried out in a standard MAC unit (including final adds in multiplications and accumulations). Carry propagation lengths in the final add and accumulation must be reduced in order to remedy this issue. PPR will be adapted to include a portion of adds (such as the last addition and accumulation), which is our primary goal. Consequently, the time required for carry propagation is minimised. The addition and accumulation of higher importance bits is not conducted until the PPR phase of the following multiplication in the suggested MAC architecture. As a result, we have two PPMs for the PPR process: one obtained from the PPG, and one produced from accumulation. The 4-bit MAC (shown in Fig. 1) is used as an example here. Our PPM is made up of two PPMs, one from the PPG the other from the accumulation, as illustrated in Fig. 1(a). Dadda trees are then used to decrease the PPM to two rows, as seen in Fig. 1(b).

Fig.1. OUR 4-BIT MAC



Due to their nonzero output for zero input, the suggested compressors have a significant impact on the mean relative error (MRE) as will be detailed later. The recommended design in this brief addresses the current issue. This improves accuracy. Based on the first bit of the operands, the static segment multiplier (SSM) produces m-bit segments from n-bit operands. Then, instead of doing n n multiplication, m m multiplication is used, where mn. Starting from the jth position, the partial product perforation (PPP) multiplicator in omits consecutive partial products starting from [0,n-1] and k is between [1, min(n-j, n-1)]. Modifying one entry inside the Karnaugh map to get a 2x2 approximate multiplier is utilised to produce 4x4 and 8x8 multipliers in this paper. For a performant Wallace tree multiplier, an incorrect counter layout has been presented. A novel approximation adder is introduced that may be used for the accumulation of the multiplier's partial products. Compared to an accurate multiplier, a 16-bit approximation multiplier achieves a 26% decrease in power.

EXISTING METHOD:

In this section, we present the proposed two-stage (i.e., two cycle) MAC architecture. The first stage performs the PPG process, the PPR process (based on the PPM that combines the PPG result and the accumulation result), the

(2N-k-1)-bit addition (i.e., a part of the final addition) and the α bit addition (for dealing with the overflow in the PPR process). Then, the second stage performs the (k+α)-bit addition to produce the accumulation result. The main features of the proposed architecture are below. λ To reduce the lengths of carry propagations, we integrate a part of additions into the PPR process. λ To handle overflow in the PPR process, a α-bit adder is used to count the total number of carries. λ By applying the gating technique, the second stage can only be executed in the last cycle (of the entire sequence of multiply-accumulate operations) for power saving. The proposed two-stage pipeline MAC unit is displayed in Fig. 5. Our PPM (for the PPR process) is composed of two PPMs: one PPM is derived by the PPG and the other PPM is derived by the accumulation. For an unsigned MAC unit, in the PPG process, "AND" gates can be directly used to generate the PPM. For a signed MAC unit, because the influences of the sign bit should be taken into account, several PPG algorithms have been proposed to generate the signed PPM. In the proposed architecture, the Baugh-Wooley algorithm is adopted in the PPG process to generate the signed PPM.

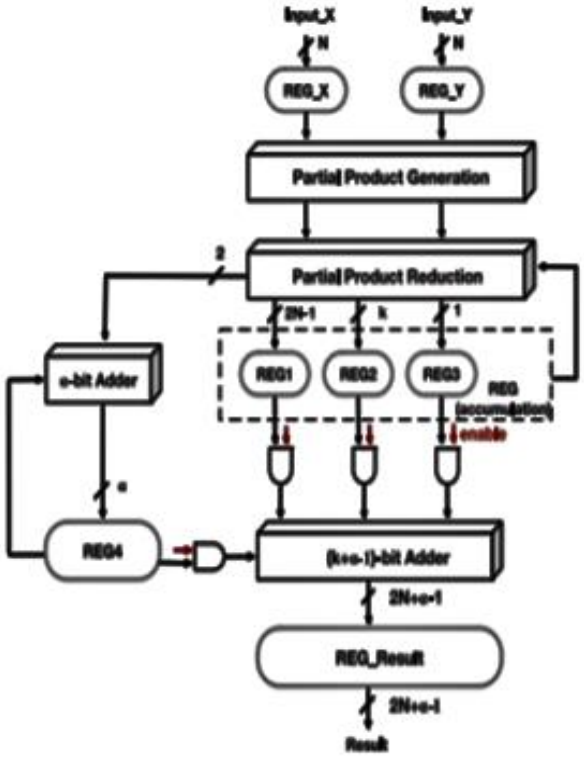


Fig .The proposed MAC architecture

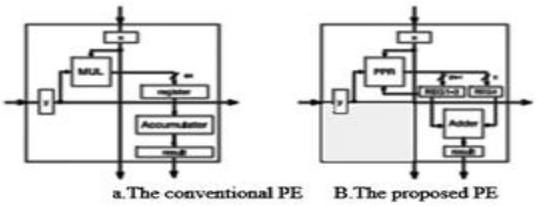
We have implemented a tool (a C++ program) to automatically generate the proposed N-bit MAC in Verilog RTL description. The users can specify the value of N and the value of k for automatic generation, where k denotes the number of higher significance bits whose additions (accumulation) are not performed in the final addition. Note that the value of k is equal to the bit width of register REG2. In our experiments, we specify the value of N to be 16

(i.e., 16-bit MAC). Besides, we assume that the maximum number of multiplications in each multiply-accumulate operation is 256.

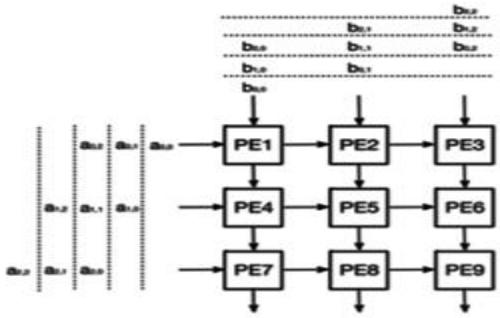
Thus, the number of guard bits (i.e., the value of α) is set to be 8. We have implemented several different configurations of the proposed MAC architecture. For the convenience of presentation, we use the term Ours_k for the naming of each configuration, where k represents the bit width of register REG2. In our experiments, these Verilog RTL descriptions are synthesized to gate-level netlists and targeted to TSMC 40 nm cell library by using Synopsys Design Compiler.

For comparisons, we also implemented the following two MAC architectures: the conventional MAC architecture and the state-of-the-art MAC architecture. In the conventional MAC architecture, the MAC unit is composed of two individual blocks (i.e., a multiplier and an accumulator). On the other hand, in the state-of-the-art MAC architecture, the multiplier and the accumulator are tightly integrated (i.e., a carry-save format is sent to the accumulator without being added to only one vector).

The systolic array has been widely used in the hardware acceleration for matrix multiplication. In recent years, several research efforts have been paid to map the inference of a convolutional neural network to a systolic array. Note that a systolic array is composed of multiple processing elements (PEs). Each PE corresponds to a MAC unit. In this section, we address the application of the proposed MAC architecture to a systolic array. Figure gives the block diagram of the PE based on the conventional MAC architecture. Note that the PE is a two-stage (i.e., two-cycle) pipeline design. The inputs of the PE are x and y. The block MUL denotes the multiplier. In the first stage, the multiplier performs the multiplication. Then, the output of the multiplier is stored in a register. In the second stage, the accumulator performs the accumulation. Then, the accumulation result is stored in register result.



a.The conventional PE B.The proposed PE



Block diagram of 3x3 systolic array

For the systolic array based on the conventional PE (i.e., the conventional MAC architecture), Table V displays the detailed operations of each PE (PE1 ~ PE9) in each cycle. In Table V, the term MUL denotes the multiplication operation (i.e., the first stage of a conventional multiply accumulate operation) and the term ACC denotes the accumulation operation (i.e., the second stage of a conventional multiply accumulate operation).

PROPOSED METHOD:

In data transmission applications, the widely used public-key cryptosystem is a simple and efficient Montgomery multiplication algorithm such that the low-cost and high-performance. In which includes encryption and decryption process. The Montgomery 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. 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. When modular multiplier is done with CCSA technique and it has some drawbacks. The drawbacks are short critical path, high power consumption. To overcome the drawbacks the CCSA is replaced with PASTA (Parallel Self Timed Adder) in the Montgomery modular multiplier. The PASTA adder can achieve less power consumption.

Modular Multiplication is the central operation in many application areas including public key cryptography for encryption and decryption. The widely used method for modular multiplication is Montgomery modular multiplier. In which there will be a carry save adder . X´Y mod M is the operation to be performed. In which X and Y are the inputs. It is necessary to find the value of mod M, henceforth going for this algorithm. Comparing all previously occurring algorithms, this algorithm will produce the optimized output. There are two cases, semi carry save addition and full carry save addition. In this semi carry save addition, the given inputs are in binary and the inter outputs alone in carry save. Whereas in full carry addition, both inputs and inter outputs are in carry save. On comparing, it can be seen that semi carry save is the most advantageous one because it has only one carry save and hence it has less area and high speed which is required for designing an VLSI based multipliers.

Consider the modulus N to be a k-bit odd number and an extra factor R is to be defined as 2k mod N, where $2k-1 \leq N < 2k$. Given two integers a and b, where a, b

A = a x R (mod N)

B = b x R (mod N)

In this existing system, carry save addition with semi-carry approach is described. In which all the multiplicands are not recycled, that is whatever the multiplicand is needed to be multiplied at that time alone is used for determining the output. The carry save approach has higher benefits since it is the basic key for operating a Montgomery modular multiplier. In such a way, using this semi carry save type only one carry level adder is

implemented which may be two serial half adders or a full adder can be used based on the requirement. It thereby reduces the number of clock cycles and hence less delay. So the output will be optimized and it can be implemented using Verilog coding.

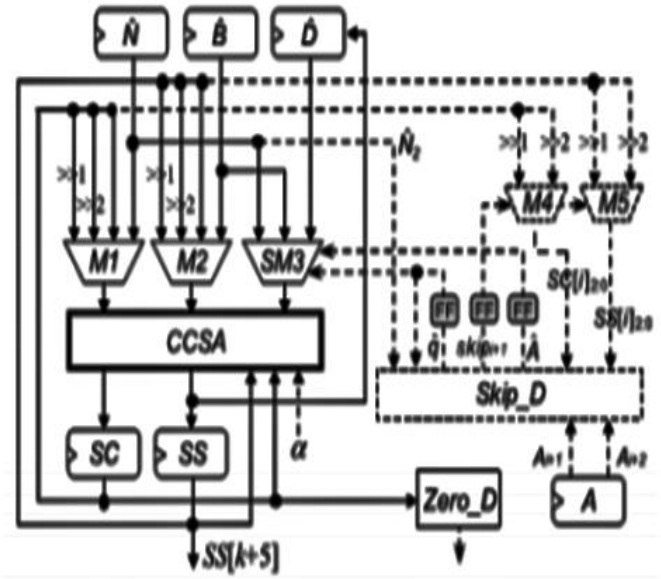


Fig. Block diagram of Montgomery Modular Multiplication using CCSA

The above architecture is the semi-carry save based Montgomery multiplier. In which the loop is reduced on comparing to the existing one. It consists of two multiplexers, one multiplier, one configurable carry save adder, flip-flops, skip detector and zero detector.

Illustrates the block diagram of proposed semi carry save multiplier. It is first used to precompute 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. 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.

Critical Path Delay Reduction: In order to reduce the critical path delay, the operations in semi carry save and full carry save is performed jointly. The carry save format conversion as well as the binary format should be taken place. Then pre-computation must be done in order to reuse the multiplicand values. Another method is using zero

detectors. SC will produce the output only when the zero is detected. Then the pre computation can be done i-1 iteration.

Clock Cycle Number Reduction: In order to decrease the clock cycle number, a configurable carry save architecture to perform one three-input carry-save addition or two serial two-input carry-save addition is used. Furthermore the number of iteration can also be reduced to reduce number of clock cycles. Then a signal skip is used. In order to verify whether i+1 is required or not to be happen in the upcoming events. This can be found in the previous i iteration itself. By again using the same condition, signal skip will use i+1, so that it increases by a factor 2. Hence it directly goes to i+2. So that clock cycle gets reduced.

Quotient Pre-computation: Ai+1, Ai+2 and qi+1, qi+2 should be known already in order that the unwanted steps in the (i +1) iteration can be reduced by determining i iteration. So as to pre compute the quotients. Another method is using skip detector so that it will pre computes the values. And also since the shortest path in this multiplier is lengthened, it has to be minimized. As modulus N is an odd number, it can be used directly for the multiplication. So that time is consumed highly.

NOTE: To increase the Speed of Operation we are replacing the CSA with PASTA (Parallel self timed adder) in the proposed architecture.

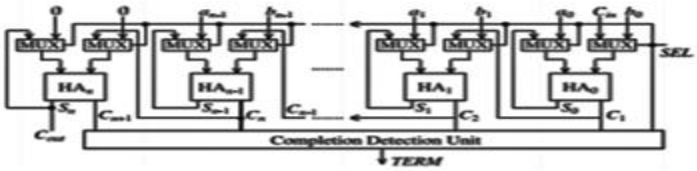


Fig.9.Block diagram of PASTA

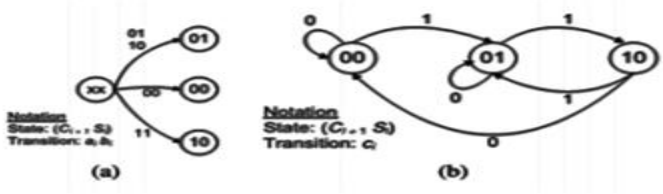


Fig. State diagram of PASTA (a).Initial phase, (b).Iterative phase

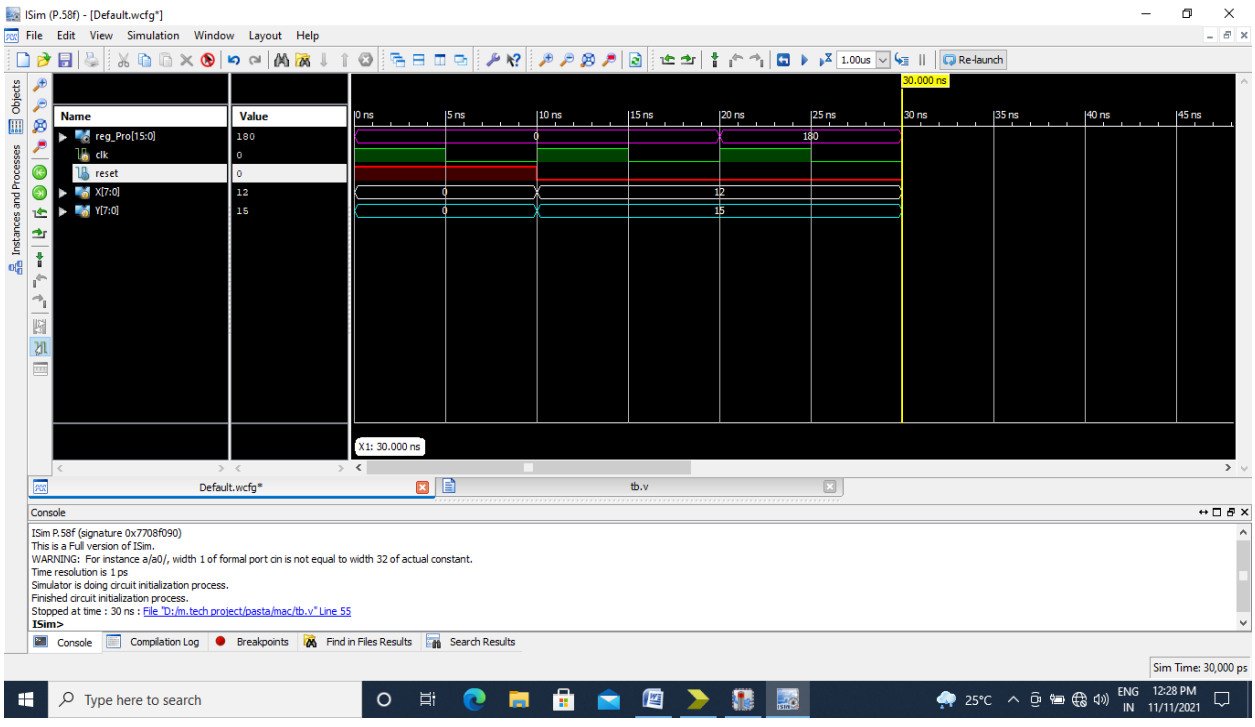
Montgomery multiplication is to perform fast modular multiplication(MM).PASTA adder using in Montgomery modular multiplication is to reduced area and clock cycles. To design a simple and efficient radix-2 Montgomery Modular multiplication with Parallel Self Timed Adder (PASTA).The design of PASTA is uses half adders (HAs) along with multiplexers requiring minimal interconnections. The selection input for two-input multiplexers corresponds to the Request handshake signal and will be a single 0 to 1 transition denoted by SEL. It will initially select the actual operands during SEL=0andwill switch to feedback/carry paths for subsequent iterations using SEL=1. The feedback path from the HAs enables the multiple iterations to continue until the completion when all carry signals will assume zero values are show in Fig .8. In Fig.9, two state diagrams are drawn for the initial phase and the iterative phase of the proposed architecture. Each state is represented by (Ci+1 Si) pair where Ci+1, Si

represent carry out and sum values, respectively, from the ith bit adder block. During the initial phase, the circuit merely works as a combinational HA operating in fundamental mode. It is apparent that due to the use of HAs instead of FAs, state (11) cannot appear.

The proposed architecture of Montgomery Modular Multiplication using PASTA adder, which consists of one one-level Parallel Self Timed Adder(PASTA) architecture, two 4-to-1 multiplexers (M1 and M2) one simplified multiplier SM3, one skip detector Skip_D, one zero detector Zero_D, and six registers. Zero detector Zero_D is used to detect SC is equal to zero. The Skip_D is composed of four XOR gates, three AND gates, one NOR gate, and two 2-to-1 multiplexers the skip detector is used to detect the unnecessary multiplication operations.

The design has been implemented using Xilinx Verilog coding. For further verification, the design can be done using Cadence. It can be clearly understand by the waveform shown below. It can be proven that it has reduced area complexity and speed complexity on comparing to all other multipliers. The method has been implemented using a configurable carry save adder so as to prove the maximum delay to be less comparing all. The delay and area can be minimized as much as possible as comparing to all other previous existing architectures.

Result: The design has been implemented using Xilinx Verilog coding. For further verification, the design can be done using Cadence. It can be clearly understand by the waveform shown below. It can be proven that it has reduced area complexity and speed complexity on comparing to all other multipliers. The method has been implemented using a configurable carry save adder so as to prove the maximum delay to be less comparing all. The delay and area can be minimized as much as possible as comparing to all other previous existing architectures.



CONCLUSION

MAC design for real-time DSP applications that is cheap on power and high on speed is presented in this study. To simplify the PPR algorithm, we propose including certain additions (including a portion of the ultimate addition in multiplication and a portion of the addition in accumulation) as part of the process. Carry propagation delays & power dissipations are decreased as a consequence. A -bit accumulator is being used to keep track of the overall number of carries throughout the PPR process. The suggested technique has been shown to consistently operate in practise via the use of experiments. Both signed and unsigned MAC units may benefit from the MAC architecture suggested here. Note that the PPM structure and the -bit addition technique are the sole changes between the unsigned and signed MAC units. It should be noted that this new MAC design may be used for both a sinusoidal and sinusoidal arrays (for performing the matrix multiplication). The suggested MAC design, as compared to the standard systolic array based on a common PE (i.e. the typical MAC architecture), reduces circuit size and power consumption by a significant margin while maintaining the same timing restriction. When compared to FCS multipliers, SCS-based multipliers use fewer clock cycles but take up less space since they preserve the inlet and outlet operands of a Montgomery MM in their carrysave format. Carry propagation delay and additional clock cycles are drawbacks of the existing design. PASTA adder is what we use to get around the drawbacks. The PASTA adder is used in the Montgomery Modular Multiplier because of its inexpensive hardware cost, short critical path delay, and decreased number of clock cycles necessary to complete a single MM operation. MM operation.

REFERENCES

[1] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[2] E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation scheme for parallel multipliers," in Proc. 31st Asilomar Conf. Signals, Circuits Syst., Nov. 1998, pp. 1178–1182.

[3] K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, "Design of low-error fixed-width modified booth multiplier," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 522–531,May 2004.

[4] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I, Reg. Papers,vol. 57, no. 4, pp. 850–862, Apr. 2010.

[5] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans.Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[6] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim,"Energy-efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[7] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,vol. 24, no. 10, pp. 3105–3117, Oct. 2016.

[8] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4,pp. 490–501, 2011.

[9] C.-H. Lin and C. Lin, "High accuracy approximate multiplier with error correction," in Proc. IEEE 31st Int. Conf. Comput. Design, Sep. 2013,pp. 33–38.

[10] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in Proc.Conf. Exhibit. (DATE), 2014, pp. 1–4.

[11] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc.IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Oct. 2011,pp. 667–673.

[12] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 63,no. 9, pp. 1760–1771, Sep. 2013.

[13] S. Sumanet al., "Image enhancement using geometric mean filter and gamma correction for WCE iamges," in Proc. 21st Int. Conf., Neural Inf. Process. (ICONIP), 2014, pp. 276–283