🌐United International Journal of Engineering and Sciences🌐
(UIJES – A Peer-Reviewed Journal); ISSN:2582-5887 | Impact Factor:8.075(SJIF)
📓Volume 5 | Special Issue 1 | 2025 Edition
National Level Conference on “Advanced Trends in Engineering
Science & Technology” – Organized by RKCE

LOW POWER HIGH SPEED POSIT MULTIPLIER FOR DIGITAL DEVICES

MR.SK.JOHN SYDHA

Assistant Professor[1], [2345]UG scholar

K.MANOHAR[2], P.RAGHAVENDRA SWAMY[3], V.PRASAD[4], D.BAJI BABU[5]

Department of Electronics and Communication Engineering

R K College of Engineering

Vijayawada, India

Abstract—

Posit number system has been used as an alternative to IEEE floating-point number system in many applications, especially the recent popular deep learning. Its non-uniformed number distribution fits well with the data distribution of deep learning and thus can speed up the training process of deep learning. Among all the related arithmetic operations, multiplication is one of the most frequent operations used in applications. However, due to the bit-width flexibility nature of posit numbers, the hardware multiplier is usually designed with the maximum possible mantissa bit-width. As the mantissa bit-width is not always the maximum value, such multiplier design leads to high power consumption especially when the mantissa bit-width is small. In this brief, a power efficient posit multiplier architecture is proposed. The mantissa multiplier is still designed for the maximum possible bit-width, however, the whole multiplier is divided into multiple smaller multipliers. Only the required small multipliers are enabled at run-time. Those smaller multipliers are controlled by the regime bit-width which can be used to determine the mantissa bit-width. This design technique is applied to 8-bit, 16-bit, and 32-bit posit formats in this brief and an average of 16% power reduction can be achieved with negligible area and timing overhead.

Index Terms—Posit number system, posit multiplier, computer arithmetic, low-power arithmetic circuit.

I.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 Daddy 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).

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.

Voltage over-scaling (VOS) may be used to approximate an 8-bit Wallace tree multiplier. An error occurs when the supply voltage is reduced, resulting in routes that do not match delay restrictions. Before now, researchers have focused on applying approximation adders and compressor to the partial products to reduce the complexity of logical expressions. Various probabilities are included into the partial products in this short. Systematic approximation is used to examine the likelihood statistics of the changing partial products.

Simplified arithmetic units are offered for approximation (the half-adder, full-adder, and 4-2 compression). The complexity of the arithmetic units has been lowered, but the error value has also been taken into consideration. Reducing the logic cost of approximate computation saves power and space, while yet allowing for higher precision. Compared to current multiplier designs, the new multipliers are more efficient in terms of size, power consumption, and inaccuracy, as well as achieving higher PSNR (peak signal to noise ratio) values in image processing. To put it another way, the arithmetic error distance (ED) is the difference between a correct result and an approximation output given the same input.

II.LITERATURE SURVEY

2.1. Approximate Adders for approximate multiplication:

Document It's becoming more difficult for CMOS technology to keep up with the demands of future applications. It's possible to narrow this gap greatly using a number of interesting design methods. It's one of them, and it's received the most attention from the scientific community in recent years. Accuracy is sacrificed for speed and energy efficiency in approximation computing, which harnesses the inherent mistake of applications and provides high-performance energy-efficient software / hardware implementations (e.g., performance and energy). Many research projects have investigated approximation computing at various levels of the computing stack during the last decade, although much of the effort at hardware abstraction has focused on adders. There is a comparative study of the most current approximation adders. It also compares design metrics based on both traditional and approximation computing design metrics.

2.2. Compressors for Multiplication:

At the nanometer scale, approximate computing is a promising approach to digital processing. For computer arithmetic designs, inexact computing is of special relevance. For a multiplier, the design and analysis of two novel approximate 4-2 compressors are discussed. When it comes to circuit-based figures of merit, these designs use various compression characteristics to ensure that imprecision in calculation (measured in terms of the error rate and the normalised error distance) can satisfy (number of transistors, delay and power consumption). A Daddy multiplier uses four distinct strategies to make use of the suggested approximation compressors. Applicability of the approximation multipliers to image processing is shown in a series of simulations. There is a considerable decrease in energy dissipation, latency and transistor count compared to an accurate design, and two of the suggested multiplier designs have good capabilities for picture multiplication in terms of average normalised error length and maximum signal-to-noise ratio (more than 50dB for the considered image examples).

2.3. Wallace-Booth Multiplier:

Recent emphasis has been given to approximate or inexact computing because of its promise for high performance and low power consumption. There are three parts to this approximation multiplier: a Booth encoder, a 4-2 compressor, and an approximate tree structure. For 8x8, 16x16, and 32x32-bit signed multiplication schemes, the approximate design is built and tested. The findings of 45 nm technology simulations are presented and analysed in this paper. For example, compared to an accurate Wallace-Booth multiplier or any other approximate multiplier known in the technical literature, our suggested approximation approach delivers considerable advantages in power consumption, latency, and other metrics. The suggested design has been shown to be viable by these outcomes.

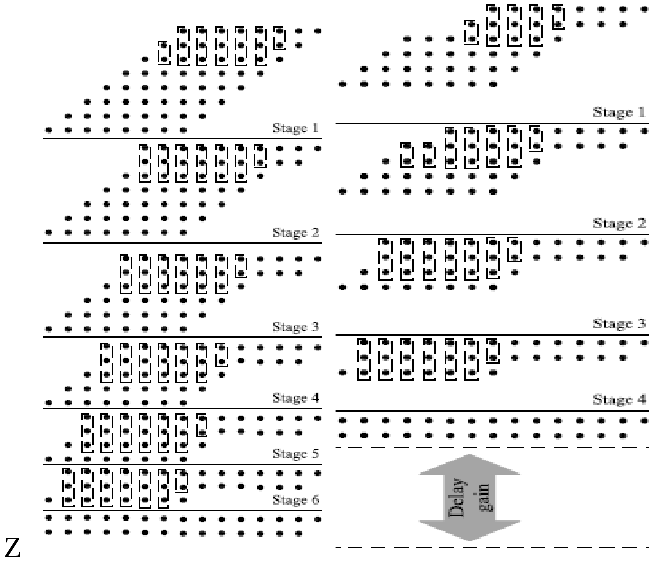
2.4. Two variants of multipliers:

For error-resistant applications, approximate computation may reduce the design complexity while increasing performance and power efficiency. A novel way to approximating multipliers in design is presented. Changes are made to the multiplier's partial products to incorporate changing probability terms. The difficulty of approximation is influenced by the possibility of accumulating changing partial products. Two 16-bit multiplier variations use the suggested approximation. Two suggested multipliers have been shown to save 72 percent and 38 percent of the power of an exact multiplier, according to the synthesis. Existing approximation multipliers can't compete with their accuracy.

Image processing is used to test the suggested multipliers, and one model gets the best peak signal to noise ratio of all of them.

III.EXISTING METHOD

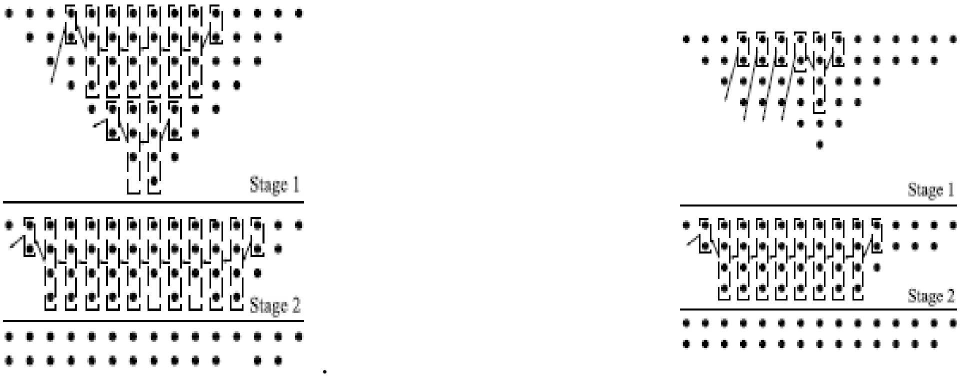
As a follow-up, a quick explanation of the estimated wallace multiplier is provided. The only difference between this multiplier and an array multiplier is the construction of the multiplier. In Fig.3(a) and 3(b), the wallace multiplier is depicted in its accurate and approximative forms. According to the illustration, the three-point combination symbolises an adder's full capacity, whereas two-point combination represents a half-adder



2(a) 2(b)

Fig-2: (a) accurate array multiplier (b) approximate array multiplier

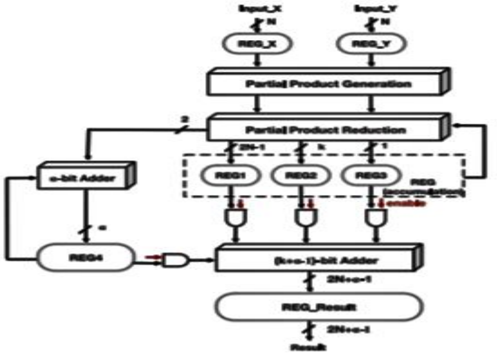
Finally, the accurate and approximate dadda multiplier by using 4:2 compressor dot diagrams is shown in Fig. 4(a) and 4(b). The combination of four dots represents the 4:2 compressor operations



4(a) 4(b)

Fig-4: (a) accurate dadda multiplier 4 (b) approximate dadda multiplier.

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 αbit 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.



Fig(5).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.

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.

VI. 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 ≤ 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.

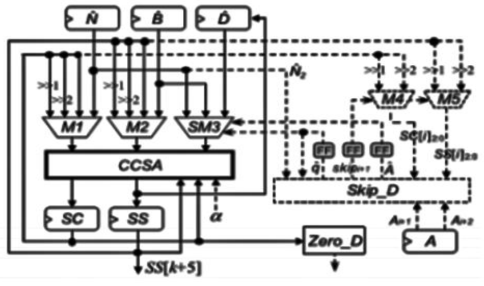


Fig.8. Block diagram of Montgomery Modular Multiplication using CCSA

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 recomputed the four-to-two carry save additions. Then the required multiplication can be performed. The modulus N and inputs will be allowed inside the twomultiplexers. 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.

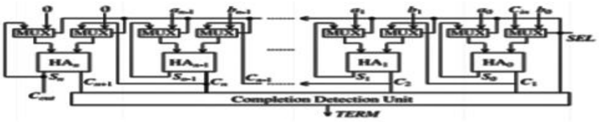


Fig.9.Block diagram of PASTA

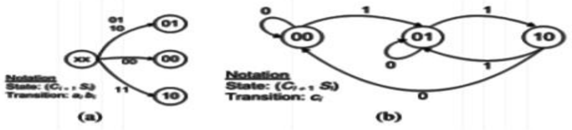
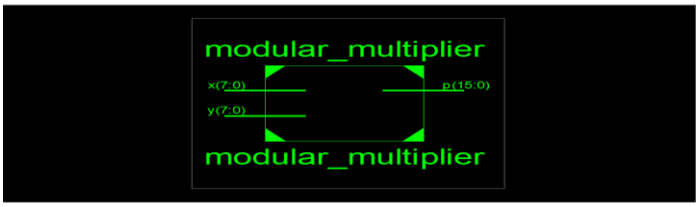


Fig.10.State diagram of PASTA (a).Initial phase, (b).Iterative phase

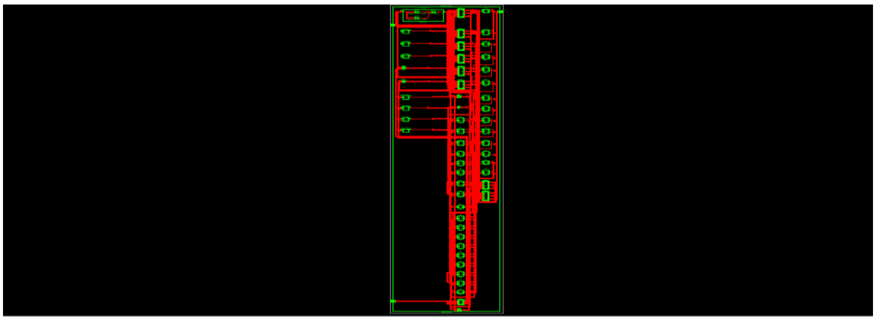
Montgomery multiplication is to perform fast modular multiplication (MM).PASTA adder using in Montgomery modular multiplication is to reduced area and clock cycles. To design a simple and efficient radix-2 Montgomery Modular multiplication with Parallel Self Timed Adder (PASTA).The design of PASTA is 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 represents carry out and sum values, respectively, from the pith 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.

V.ANALYSIS OF RESULTS

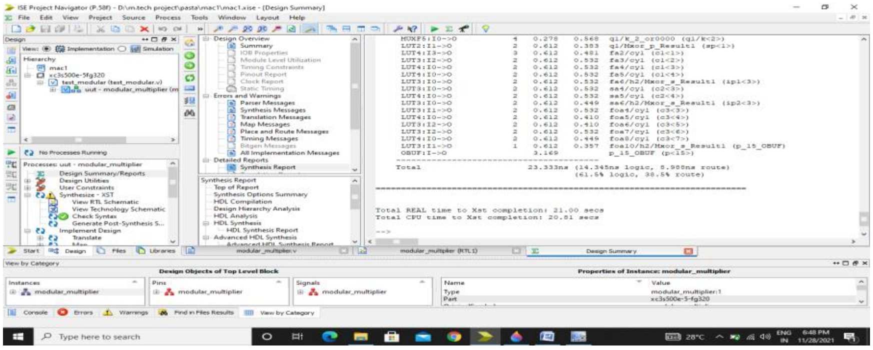
The design has been implemented using Xilinx Verilog coding. For further verification, the design can be done using Cadence. It can be clearly understand by the waveform shown below. It can be proven that it has reduced area complexity and speed complexity on comparing to all other multipliers. The method 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.



5.1 SCHEMATIC DIAGRAM:



5.2 TIME DELAY:

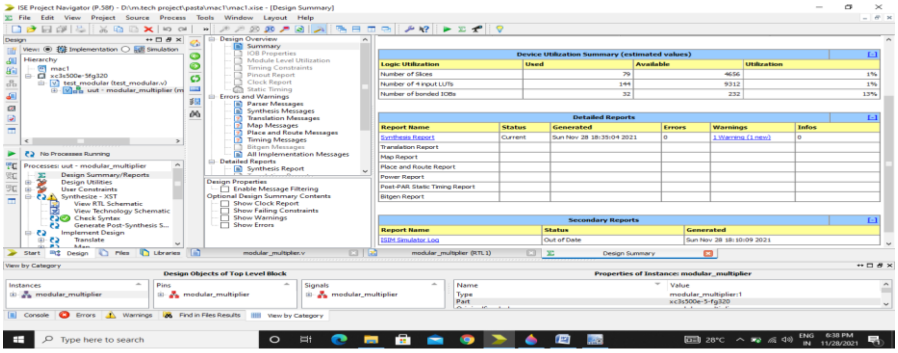


```
   MUXF5:I0->O          4   0.278   0.568  q1/k_2_or0000 (q1/k<2>)
   LUT2:I1->O           2   0.612   0.383  q1/Mxor_p_Result1 (sp<1>)
   LUT4:I3->O           3   0.612   0.481  fa2/cy1 (c1<1>)
   LUT3:I2->O           2   0.612   0.532  fa3/cy1 (c1<2>)
   LUT4:I0->O           2   0.612   0.532  fa4/cy1 (c1<3>)
   LUT4:I0->O           2   0.612   0.532  fa5/cy1 (c1<4>)
   LUT3:I0->O           2   0.612   0.532  fa6/h2/Mxor_s_Result1 (ip1<3>)
   LUT3:I0->O           2   0.612   0.532  sa4/cy1 (c2<3>)
   LUT4:I0->O           2   0.612   0.532  sa5/cy1 (c2<4>)
   LUT3:I0->O           2   0.612   0.449  sa6/h2/Mxor_s_Result1 (ip2<3>)
   LUT3:I1->O           2   0.612   0.532  foa4/cy1 (c3<3>)
   LUT4:I0->O           2   0.612   0.410  foa5/cy1 (c3<4>)
   LUT3:I2->O           2   0.612   0.410  foa6/cy1 (c3<5>)
   LUT3:I2->O           2   0.612   0.532  foa7/cy1 (c3<6>)
   LUT4:I0->O           2   0.612   0.449  foa8/cy1 (c3<7>)
   LUT3:I1->O           1   0.612   0.357  foa10/h2/Mxor_s_Result1 (p_15_OBUF)
   OBUF:I->O                3.169          p_15_OBUF (p<15>)
  ----------------------------------------
  Total                     23.333ns (14.345ns logic, 8.988ns route)
                                     (61.5% logic, 38.5% route)

=========================================================================

Total REAL time to Xst completion: 21.00 secs
Total CPU time to Xst completion: 20.81 secs

-->
```

Properties of Instance: modular_multiplier

Name	Value
Type	modular_multiplier:1
Part	xc3s500e-5-fg320

5.3 AREA CONSUMPTION :



Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	79	4656	1%
Number of 4 input LUTs	144	9312	1%
Number of bonded IOBs	32	232	13%

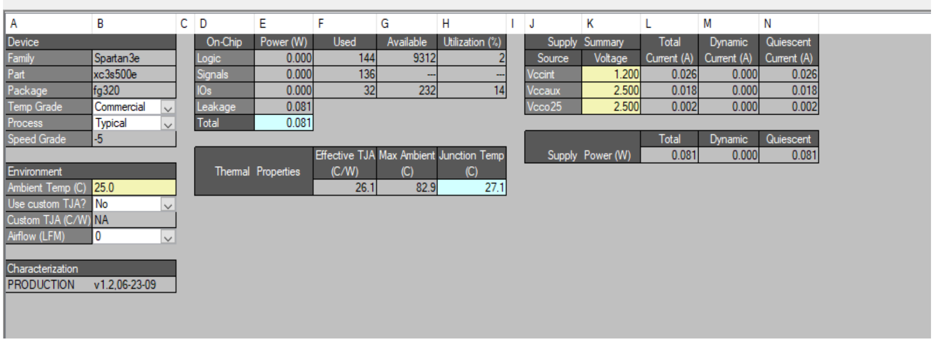
Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sun Nov 28 18:35:04 2021	0	1 Warning (1 new)	0
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports

Report Name	Status	Generated
ISIM Simulator Log	Out of Date	Sun Nov 28 18:10:09 2021

5.4 PASTA POWER CONSUMPTION:



Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Temp Grade	Commercial
Process	Typical
Speed Grade	-5

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

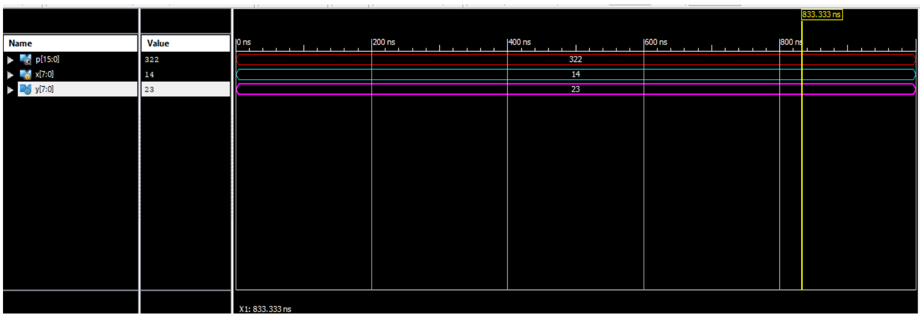
On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	144	9312	2
Signals	0.000	136	---	---
IOs	0.000	32	232	14
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.081	0.000	0.081

5.4 SIMULATION RESULT:



IV. 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.

REFERENCES

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

[2] E. J. King and E. E. Swartz lander, Jr., "Data-dependent truncationscheme 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. Paris, "Design of low-error fixed-width modified booth multiplier," IEEE Trans. VeryLarge Scale integer. (VLSI) Syst., vol. 12, no. 5, pp. 522–531,May 2004.

[4] H. R. Maharani, A. Hamada, S. M. Fakhraie, and C. Lucas, "Bio-inspiredimprecise 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. Mokena, J. Han, P. Montuschi, and F. Lombardi, "Design andanalysis 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 processingand classification applications," IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.
[7] G. Savakis, K. Tsunamis, S. Xydis, D. Souris, and K. Pekmestzi,"Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large Scale Integra. (VLSI) Cyst. Vole. 24, no. 10, pp. 3105–3117, Oct. 2016.
[8] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy firepower 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. compote. Design, Sep. 2013,pp. 33–38.
[10] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performanceapproximate multiplier with configurable partial error recovery," in Proc.Conf. Exhibit. (DATE), 2014, pp. 1–4.
[11] R. Venkatesan, A. Gadwall, K. Roy, and A. Raghunathan, "MACACO:Modeling and analysis of circuits for approximate computing," in Proc.IEEE/ACM Int. Conf. computed.-Aided Design (ICCAD), Oct. 2011,pp. 667–673.
[12] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability ofapproximate 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 andgamma correction for WCE iamges," in Proc. 21st Int. Conf., NeuralInf. Process. (ICONIP), 2014, pp. 276–283.