Design And Analysis Of Efficient Of 32 Bit Approximate Multiplier Compressors

D.Sathya[1],P.Rajesh Kumar[2]
[1]Dept of ECE
[2]Assistant Professor, Dept of ECE
[1, 2] Tagore Institute Of Engineering And Technology(Tamilnadu)

Abstract- *Approximate computing has become an emerging technique to reduce power consumption. In numerous applications, multiplication is a crucial operation, designing it for approximation is effective in optimizing system performance. In this paper, we propose low-power FPGA-based multipliers by employing the novel compressor designs. Given that the compressor is the primary unit in the multiplier, we introduce novel exact and approximate compressors with low-complexity circuits to parallels accumulate the elements. To flexibly configure the proposed compressors, a compressor-based once-through structure is proposed to 8×8 multipliers. Two variants of the approximate multipliers are provided with different accuracy hardware trade-offs. Compared with the exact multiplier, the proposed approximate multiplier reduces power by 57.90%, area by 33.80%, and delay by 24.78%. With a similar accuracy loss, the proposed designs save more hardware resources than others. In addition, the effectiveness of approximate multipliers is assessed in image sharpening.*

Keywords- Approximate computing, FPGA-based compressor, Low-power circuit.

I. INTRODUCTION

Approximate computing, which trades computation precision for power optimization, is an appealing choice for error-tolerant applications like multimedia processing and machine learning. These applications heavily depend on multiplication operations, thus designing the approximate multiplier is an effective approach to reduce the power consumption of applications and systems. Generally, multiplication involves three phases: partial product generation, partial product accumulation, and final addition. The partial product accumulation phase typically leads to primary hardware consumption caused by the significant utilization of compressors. Therefore, an approximate compressor design is important to optimize the whole multiplier performance. The state-of-the-art approximate multipliers mainly focus on ASIC-based implementations and are designed by approximate compressors with fewer logic gates.

Nowadays, with the growing popularity of FPGAs, research on FPGA-targeted approximate multipliers is more and more attractive. There are two main reasons. Firstly, because of the architectural difference between ASICs and FPGAs, the abundant researches in ASIC-based approximate multipliers usually offer asymmetrical gains in FPGA-based accelerators. As reported in, ASIC-based designs with gains of energy-delay-product (EDP) more than 30% over the exact multiplier, while only about 10% EDP saving after FPGA-based implementation. Secondly, FPGAs have the superiorities of reconfiguration capability, high energy efficiency, and fast development cycle. More and more multimedia applications tend to use FPGAs to process data. Despite the high performance of multiplication operations provided by DSP blocks in FPGAs, the designs using logic-based soft multipliers are indispensable. Thus, Xilinx and Intel also provide logic-based soft multipliers. Based on these factors, designing approximate multipliers with FPGA fabrics is desirable.

In, an approximate 4×2 multiplier is introduced by utilizing four LUTs and used to construct 4×4 and 8×8 multipliers. In, three approximate units based on LUTs are proposed and applied in the partial product matrix (PPM), generating three approximate multipliers. In, three types of 4×4 approximate multipliers are designed, and a wide range of approximate 8×8 multipliers are proposed using 4×4 multipliers. In, an exact 4×4 multiplier is proposed, and three approximate multipliers are designed by reorganizing LUTs. Most of the existing works on FPGA-based approximate multipliers save hardware resources. However, there still are two challenges for low-power designs. One issue is the absence of approximate compressor design in FPGA-based multipliers. Considering that the compressor is a power-intensive unit, its approximate design can effectively reduce power. However, because of the port limitations of FPGA fabrics, there are few studies on FPGA-based multipliers with approximate compressors. Another concern is the decreasing effectiveness of power savings resulting from the recursive construction method. Building large-size multipliers from small multipliers is a usual method in previous works. Nevertheless, as the input width increases, there is a corresponding increase in circuit complexity. These above

issues restrict the effectiveness of approximate computing for low-power multiplier designs.

In this paper, we propose low-power FPGA-based multipliers by focusing on the compressor designs. To effectively configure compressors, a compressor-based once-through structure is introduced, which differs from the traditional recursive construction method in previous works. Our contributions are as follows:

A novel exact compressor is proposed by lagging calculation of the carry result of partial products. The input ports of the LUT can be fully utilized by this carry-lagged compressor. An approximate :2 compressor is introduced to aggregate several elements into two equal-weight vectors. Additionally, a uniform expression for the proposed m:2 compressor is provided to facilitate the expanded usage of the approximate compressor.

A once-through methodology of the approximate 8×8 multiplier is presented by utilizing the proposed compressors on the partial product accumulation. Two variants of approximate 8×8 multipliers are is presented by utilizing the proposed compressors on the partial product accumulation. Two variants of approximate 8×8 multipliers are introduced to meet varying performance requirements.

II. LITERATURE REVIEW

2.1 DESIGN AND ANALYSIS OF APPROXIMATE REDUNDANT BINARY MULTIPLIERS

As technology scaling is reaching its limits, new approaches have been proposed for computational efficiency. Approximate computing is a promising technique for high performance and low power circuits as used in error-tolerant applications. Among approximate circuits, approximate arithmetic designs have attracted significant research interest. In this paper, the design of approximate redundant binary (RB) multipliers is studied. Two approximate Booth encoders and two RB 4:2 compressors based on RB (full and half) adders are proposed for the RB multipliers. The approximate design of the RB-Normal Binary (NB) converter in the RB multiplier is also studied by considering the error characteristics of both the approximate Booth encoders and the RB compressors. Both approximate and exact regular partial product arrays are used in the approximate RB multipliers to meet different accuracy requirements. Error analysis and hardware simulation results are provided. The proposed approximate RB multipliers are compared with previous approximate Booth multipliers; the results show that the approximate RB multipliers are better than approximate NB Booth multipliers especially when the word size is large. Case studies of error-resilient applications are also presented to show the validity of the proposed designs.

2.2. LOW-POWER COMPRESSOR- BASED APPROXIMATE MULTIPLIERS WITH ERROR CORRECTING MODULE,"

This letter proposes an unsigned approximate multiplier architecture segmented into three portions: the least significant portion that contributes least to the partial product (PP) is replaced with a new constant compensation term to improve hardware savings without sacrificing accuracy. The PPs in the middle portion are simplified using a new 4:2 approximate compressor, and the error due to approximation is compensated using a simple yet efficient error correction module. The most significant portion of the multiplier is implemented using exact logic as approximating it will results in a large error. Experimental results of 8-bit multiplier show that the power and power-delay products are reduced up to 47.7% and 55.2%, respectively, in comparison with the exact design and 36.9% and 39.5%, respectively, in comparison with the existing designs without significant compromise on accuracy.

III. METHODOLOGY

APPROXIMATE MULTIPLIER

A brief description about Approximate Multiplier is given in this section used to data encoded using LUT.

3.1 CALCULATION OF PARTIAL PRODUCTS

Each of the 8-bit partial products is implemented by a 74284/74285 pair. The subsystem has 16 inputs, the multiplicand and multiplier, and 32 outputs, constituting the four 8-bit partial products. The partial product subsystem is shown in figure The low-order 4 bits of the final product, P3-0, are the same as PP03-0 and do not participate in the sums. P7-4 and P11-8 are sums of three 4-bit quantities. To cascade full adders to implement a function that sums three 4-bit quantities, denoted A3-0, B3-0, and C3-0. (Watch that you don't confuse the variable Ci with adder carry-ins.) The first level of full adders sums 1 bit from each of the three numbers to be added. To accomplish this by using the carry input as a data input. The second-level adders combine the carry- out from the next lower order stage with the sum from the first-level adder. The carries simply propagate from right to left among the second-level adders.

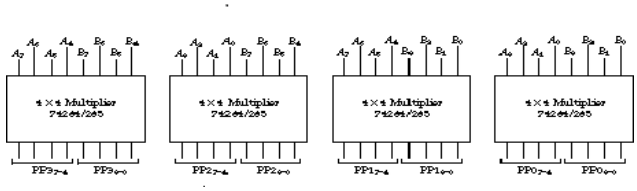

Figure1: Partial product subsystem

The first-level full adders are provided by 74183 dual binary adders. The second-level adders are implemented by a 74181 arithmetic logic unit, configured for the adder function. The extra performance advantage of internal carry look-ahead logic. Note that the ALU block is written in its positive logic form, with positive logic data inputs and outputs andnegative logic carry-in and carry-out. It provides the basic building block we can use to implement bit slices P7-4 and P11-8 for the result products.

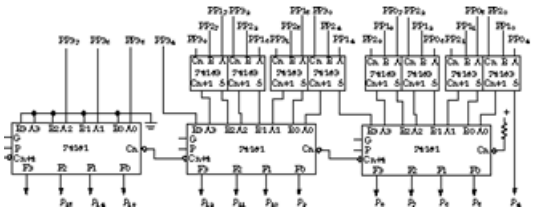

Figure2 :Basic building block bit slices P7-4 and P11-8

The shift-add Multiplier scheme is the most basic of unsigned Integer multiplication algorithms. The operation of multiplication is rather simple in digital electronics. It has its origin from the classical algorithm for the product of two binary numbers. This algorithm uses addition and shift left operations to calculate the product of two numbers.The left example shows the multiplication procedure of two unsigned binary digits while the one on the right is for signed multiplication. The first digit is called Multiplicand and the second Multiplier. The only difference between signed and unsigned multiplication is that we have to extend the sign bit in the case of signed one, as depicted in the given right example in PP row Based upon the above procedure.Compares the two results to select a smaller one. The survivor path unit records the survivor path of each state selectedby the ACS module. Once the trellis diagram is reconstructed, tracing back through the trellis is performed.

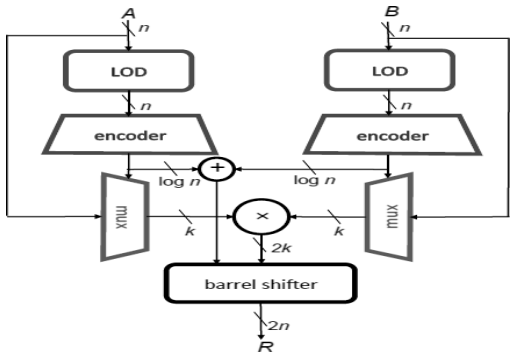

Figure 3: The simplified schematics for the proposed approximate multiplier

The main operation in the process of multiplication of two numbers is addition of the partial products. Therefore, the performance and speed of the multiplier depends on the performance of the adder that forms the core of the multiplier. To achieve higher performance, the multiplier must be pipelined. Throughput is often more critical than the cycle response in DSP designs. In this case, latency in the multiply operation is the price for a faster clock rate. This is accomplished in a multiplier by breaking the carry chain and inserting flip-flops at strategic locations. Care must be taken that all inputs to the adder are created by signals at the same stage of the pipeline. Delay at this point is referred to as latency.

Our proposed approximate multiplier exploits the fact thatnot all bits of a number are equally important. Thus, wepropose to limit the number of bits applied to the multiplierby carefully selecting a range of bits for each of the twooperands of the multipliers. Therefore, in hardware cost, ourapproach reduces a large multiplier to a significantly smallercore multiplier and some steering logic that is required todetect and route the selected range of bits of each of theoperands to the smaller core multiplier.In our design, we propose a dynamic and fast bit selectionscheme to reduce the size of the multiplier while introducing bounded unbiased error to the multiplication result. Assumingeach operand has n bits, our design uses two leading onedetector (LOD) circuit blocks to dynamically locate the most significant “1” in each of the two operands as illustrated in Figure For each operand, the location of the mostsignificant “1” is then used to select the following k_2 consecutivenumber of bits based on the required accuracy. Here,k is a designer-defined value which specifies the bandwidthused in the core accurate multiplier.

Handling of signed numbers: Due to higher utilizationof unsigned multiplication in applications most amenable toapproximation, i.e. image processing and machine vision,proposed arithmetic units in approximate computing have beenlargely focused on unsigned multiplication. Our

proposed approximatedesign can support signed operation with a straightforwardextension. For this purpose, pre-processing logic canbe added in order to convert the signed operands to unsignedinputs using twos complement before forwarding them to anunsigned approximate block. The sign signal for the result isthen calculated separately and the output will be negated ifnecessary.

1V. SIMULATION RESULTS

4.1 RTL SCHEMATIC

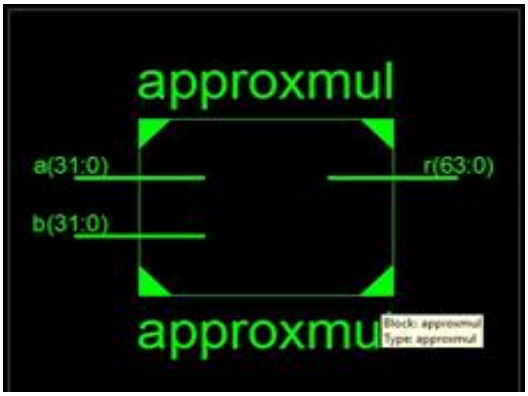

Figure 4: RTL View of Approximate Multiplier

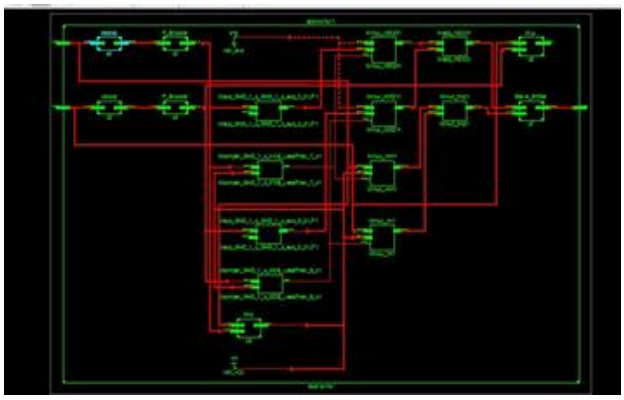

Figure 5: Technology View of Approximate Multiplier

Figure 6: Xilinx Output

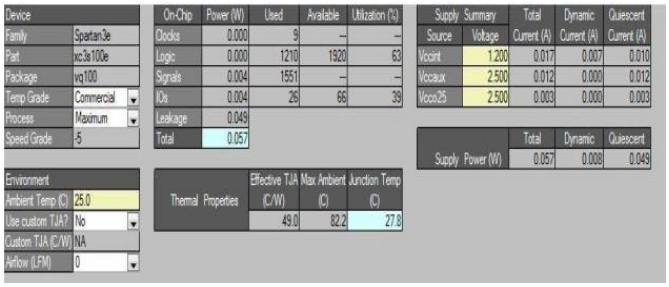

Figure 9: Power Analysis of Approximate Multiplier

V. CONCLUSION

In this paper, the power-efficient FPGA-based multipliers are proposed by using approximate computing concept. Firstly, a carry-lagged compressor is proposed to compress the partial products exactly. To compress the remaining elements into 2 rows, an approximate m:2 compressor is designed. Secondly, two variants of approximate 8×8 multipliers with different accuracy are provided by using both types of compressors in a once-through structure. Finally, the experimental results indicate that the proposed multipliers achieve more hardware savings compared with other designs with similar accuracy.

REFERENCES

[1] W. Liu, T. Cao, P. Yin, Y. Zhu, C. Wang, E. E. Swartzlander, and F. Lo mbardi, “Design and analysis of approximate redundant binary multipliers”, IEEE Transactions on computers, vol. 68, no. 6, pp. 804-819, 2019.

[2] U. A. Kumar, S. K. Chatterjee, and S. E. Ahmed, “Low-Power Compressor- Based Approximate Multipliers with Error Correcting Module,” IEEE Embedded Systems Letters, vol. 14, no. 2, pp. 59-62, 2022.

[3] M. Zhang, S. Nishizawa and S. Kimura, “Area Efficient Approximate 4-2 Compressor and Probability-Based Error Adjustment for Approximate Multiplier,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 70, no. 5, pp. 1714-1718, 2023.

[4] H. Pei, X. Yi, H. Zhou, and Y. He, “Design of Ultra-Low Power Consumption Approximate 4-2 Compressors Based on the Compensation Characteristic,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 1, pp. 461-465, 2020.

[5] B. S. Prabakaran, V. Mrazek, Z. Vasicek, L. Sekanina, and M. Shafique, “ApproxFPGAs: Embracing ASIC-Based Approximate Arithmetic Components for FPGA-Based Systems,” in Design Automation Conference (DAC), pp. 1-6, 2020.

[6] S. Ullah, S. Rehman and et al., “Area-Optimized Low-Latency Approximate Multipliers for FPGA-based Hardware Accelerators,” in Design Automation Conference (DAC), pp. 1-6, 2018.