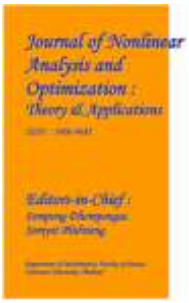
**J**ournal of **N**onlinear **A**nalysis and **O**ptimization

Vol. 15, Issue. 1, No.15 : 2024

ISSN : **1906-9685**



# Design of Approximate Multiplier Using Approximate High-Order Compressors

**Thumu.Satyanarayana**[1] **Mr.M.Devadas**[2]

thumusatyanarayana98@gmail.com

[1]**M.TECH VLSI Vaagdevi College Of Engineering, Bollikunta, Warangal, Telangana.**

[2]**Associate Professor Vaagdevi College Of Engineering, Bollikunta, Warangal, Telangana.**

**Abstract: To reduce the power consumption, the design of approximate multiplier appears as a promising solution for many error-resilient applications. In the existing paper, a low-power high-accuracy approximate 8 x 8 multiplier design is proposed, where the power consumption can be saved with a small error. To our knowledge, the existing design is the first work that successfully uses high order approximate compressors in the approximate multiplier design. Based on the existing design a 16 x 16 multiplier is proposed in this paper. The proposed design is innovated by using an efficient adder namely Brent Kung adder and this is implemented in the FIR filter design. Synthesis and simulation of the proposed design is done in Xilinx 14.7. Keywords-approximate computing, multiplier, low-power design; innovation.**

## I. INTRODUCTION

Multiplication is a crucial fundamental arithmetic operation in digital signal processing. To reduce the power consumption of an embedded system, the design of approximate multiplier appears as a promising solution for many error-resilient applications. In, different approximate multiplier designs (based on approximate 4:2 compressors) have been proposed to save the power consumption. However, their NMED (normalized mean error distance) values are relatively large. Liu et al. limited the carry propagation to the nearest neighbors for fast partial product accumulation. Zervakis et al. used partial product perforation to reduce the power consumption. Yang et al. used carry-maskable adders to reduce the length of carry chain. These approaches rely on extra logic for post-processing (i.e., error recovery) to reduce the error distance. In this paper, we propose a low-power high-accuracy approximate 8 x 8 multiplier design. The main contribution of our work is that we successfully demonstrate that high order approximate compressors (e.g., 8:2 compressor) can be used in the approximate 8 x 8 multiplier design for achieving lower power dissipation while still maintaining high accuracy. Note that the proposed design does not require any extra logic for post-processing (i.e., error recovery). The architecture of the proposed approximate 8 x 8 multiplier design has the following two main features. (1) Significance Driven Logic Compression. According to the significance, different weights use different compressors (i.e., counters) to accumulate their product terms. The higher significance weights use accurate 4:2 compressors, the middle significance weights use near accurate compressors, and the lower significance weights use inaccurate compressors. As a result, the power consumption can be reduced with a small error. (2) High Order Approximate Compression. For the middle significance weights, we use high-order approximate compressors (e.g., 8:2 compressor) to reduce the logic of carry chains. As a result, both the delay and the power can be greatly saved. To the best of our knowledge, the proposed design is the first work that utilizes high-order approximate compressors in the approximate multiplier design. Note that this architecture allows the designers to configure the number of higher

significance weights, the number of middle significance weights and the number of lower significance weights for the trade-off between the power dissipation and the computational accuracy. Compared with an exact multiplier (Dadda tree multiplier), experimental results show that the proposed approximate multiplier can achieve 14.62% ~ 25.92% reduction in power consumption with only 0.07% ~ 0.89% NMED. Therefore, the proposed approximate multiplier does achieve both low power and high accuracy. Digital computer arithmetic is an aspect of logic design with the objective of developing appropriate algorithms in order to achieve an efficient utilization of the available hardware. Given that the hardware can only perform a relatively simple and primitive set of Boolean operations, arithmetic operations are based on a hierarchy of operations that are built upon the simple ones. Since ultimately, speed, power and chip area are the most often used measures of the efficiency of an algorithm, there is a strong link between the algorithms and technology used for its implementation. Our research aims at improving the speed of the binary multiplication by using various adders like carry look ahead adder and carry save adder. The objective of our project is to use various types of adders and compare the results in terms of delay and the power consumed in the multiplier. We are aiming at using different algorithms for multiplier such as Booths algorithm, Vedic multiplier, and so on. In order to build a multiplier which consumes less power and delivers results faster, we will be using ripple carry adder, carry look ahead adder, and carry save adder.

## II. Literature Survey

Approximate circuit design is an innovative paradigm for error-resilient image and signal processing applications. Multiplication is often a fundamental function for many of these applications. In this paper, three approximate compressors are proposed with an accuracy constraint for the partial product reduction (PPR) in a multiplier. Both approximation and truncation are considered in the approximate multiplier design. An image sharpening algorithm is then investigated as an application of the proposed multiplier designs. Extensive simulation results show that the proposed designs achieve significant reductions in area and power while achieving a high signal-to-noise ratio (SNR > 35 dB), compared to their exact counterparts as well as other approximate multipliers.

Inexact (or approximate) computing is an attractive paradigm for digital processing at nanometric scales. Inexact computing is particularly interesting for computer arithmetic designs. This paper deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. These designs rely on different features of compression, such that imprecision in computation (as measured by the error rate and the so-called normalized error distance) can meet with respect to circuit-based figures of merit of a design (number of transistors, delay and power consumption). Four different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier. Extensive simulation results are provided and an application of the approximate multipliers to image processing is presented. The results show that the proposed designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design; moreover, two of the proposed multiplier designs provide excellent capabilities for image multiplication with respect to average normalized error distance and peak signal-to-noise ratio (more than 50 dB for the considered image examples).

## III. HIGH ORDER COMPRESSORS.

The critical path of a multiplier is often related to the maximum height of PPM (partial product matrix). Thus, there is a need to compress the PPM. A n:2 compressor is a slice of a multiplier that reduces n numbers (i.e., product terms) to two numbers when properly replicated. In slice i of the multiplier, the n:2 compressor receives n bits in position i and one or more carry bits from the lower positions (such as i-1), and produces two output bits in positions i and i+1 and one or more carry bits into the higher positions.

Conventionally, 4:2 compressors are used in the multiplier design. Fig. 1(a) gives the block diagram of an accurate (i.e., exact) 4:2 compressor. The four input bits are denoted as X0, X1, X2 and X3. The two output bits in positions i and i+1 are denoted to as Sum and Carry respectively. The carry bit from the lower position is denoted as Cin while the carry bit into the higher position is denoted as Cout.

Fig. 1(b) gives the block diagram of an approximate 4:2 compressor. To save the logic of carry chains, the carry bits Cin and Cout are omitted. Moreover, to reduce the error rate, the logics of Sum and Carry are re-designed (i.e., different from the logics of Sum and Carry in an accurate 4:2 compressor).

Previous works did not consider high-order compression (i.e., did not consider n ≥ 5). In fact, high-order compression can further reduce the delay and power. In this section, we introduce our high-order compressor design (i.e., n ≥ 5).

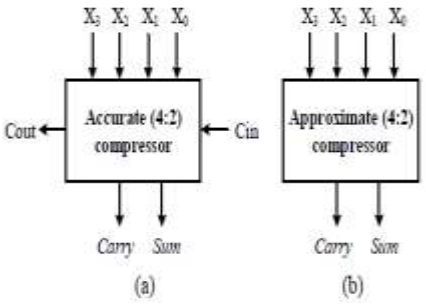



Figure 1. (a) Accurate 4:2 compressor (b) Approximate 4:2 compressor.

The Approximation of Carry Here, we study the approximation of the logic of the Carry output. In a conventional half adder, the carry bit Ch is defined as below:

$$C_h\,(X0, X1) = X0.X1$$

In a conventional full adder, the carry bit Cf is defined as below:

$$C_f(X_0,X_1,X_2) = X_0.\ X_1 + X_1.\ X_2 + X_0.\ X_2$$

As described, we can implement the equation (1) as a modified half adder, and implement the equation (2) as a modified full adder. Fig. 2 (a) and Fig. 2 (b) give the logic of modified half adder and the logic of modified full adder, respectively. Then, based on the modified half adder and the modified full adder, we can construct the approximation logic for the Carry output of a high-order approximate compressor. In the following, we use the Carry output of our approximate 5:2 compressor examples. When the number of input bits is 5 (i.e., n = 5), we can split the 5 input bits into 2 groups: one group includes X0, X1, and X2, and the other group includes X3 and X4. Then, the Carry output of our approximate 5:2 compressor is as below:

$$C_f\,(X_0,X_1,X_2) + C_h\,(X_3,X_4) + C_h\,(X_0{+}X_1{+}X_2,X_3{+}X_4).$$

Fig. 3 displays the logic of the Carry output of our approximate 5:2 compressor.

When the number of input bits is 8 (i.e., n = 8), we can split the 8 input bits into 3 groups: one group includes X0, X1, and X2, one group includes X3, X4, and X5, and one group includes X6 and X7. Then, the Carry output of our approximate 8:2 compressor is as below:

Cf (X0, X1,X2) + Cf (X3,X4,X5) + Ch(X6,X7) + Cf(X0+X1+X2, X3+X4+X5, X6+X7).

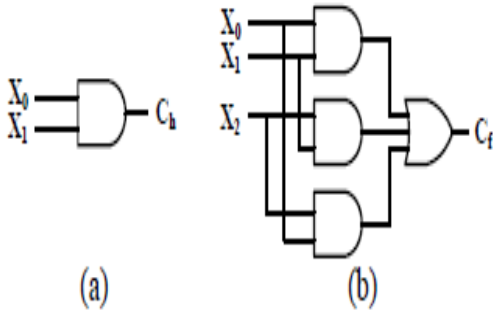



Figure 2. (a) Modified half adder (b) Modified full adder.

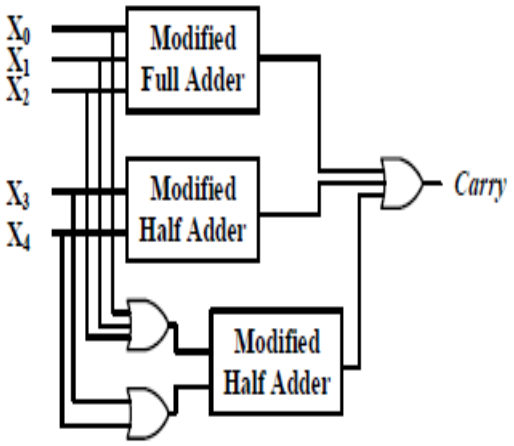



Figure 3. The logic of Carry output of our approximate 5:2 compressor.

A. The Approximation of Sum

Here, we study the approximation of the logic of Sum output. Conventionally, the tree of XOR gates are used to produce the output Sum. However, compared with other logic gates, XOR gate often has larger design overheads.

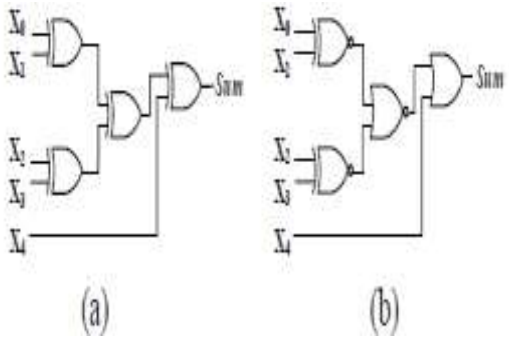



Figure 4. Sum of 5:2 compressor. (a) Accurate (b) Our approximate

We construct the approximation logic (a tree of logic gates) for the Sum output of a high-order approximate compressor as below. At the first level, we use XNOR gate instead of XOR gate. Note that the output of XNOR gate is the inverse of the output of XOR gate. To compensate for the error rate, we use NOR gate at the second level and OR gate at the third level. Since all XOR gates are replaced by other logic gates, the design overheads are greatly saved. In the following, we use the Sum output of 5:2 compressor (Fig. 4) as examples. Fig 4 (a) gives the logic of the Sum output of an accurate 5:2 compressor. Fig 4(b) gives the logic of the Sum output of our approximate 5:2 compressor.

*APPROXIMATE MULTIPLIER DESIGN*

Typically, a multiplier consists of three parts. In the first part, AND gates are utilized to generate partial products. In the second part, the maximum height of PPM (partial product matrix) is reduced by using a carry save adder tree. In the third part, a carry propagation adder is used to produce the final result. The design complexity of a multiplier is primarily related to the PPM reduction circuitry (i.e., the multiplier is primarily related to the PPM reduction circuitry (i.e., the second part). Thus, the study of multiplier design focuses on the optimization of the PPM reduction circuitry.

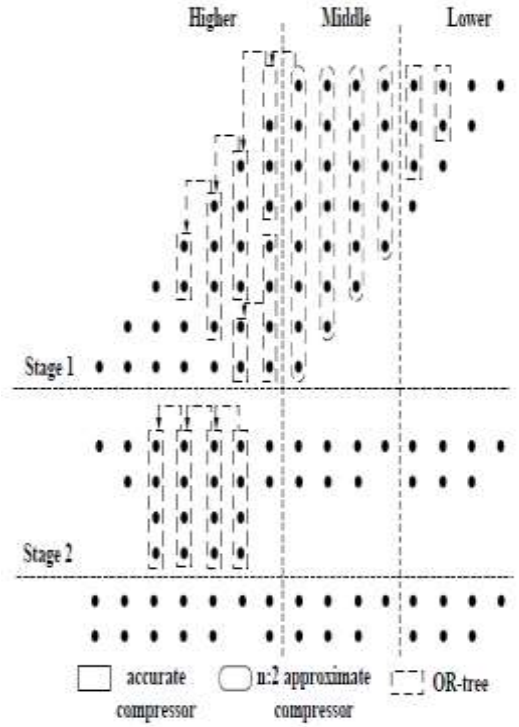



Figure 5. The PPM reduction in the existing approximate multiplier.

In this section, we propose an approximate 8 x 8 multiplier design. Fig. 6 gives the overall structure of our PPM reduction circuitry. According to the significance, the weights are classified into three categories: the higher significance weights, the middle significance weights, and the lower significance weights. Note that the designers are allowed to configure the number of higher significance weights, the number of middle significance weights and the number of lower significance weights for the trade-off between the power consumption and the computational accuracy.

To reduce the power consumption with a small error, our PPM reduction circuitry applies the significance driven logic compression technique as below: the higher significance weights use accurate (i.e., exact) 4:2 compressors; the middle significance weights use our approximate high-order compressors (i.e., the approximate n:2 compressors proposed in Section II); the lower significance weights use inaccurate compressors (OR-tree based approximation).

Our PPM reduction circuitry has two stages. The first stage is for all the weights. The second stage is only for the higher significance weights. After the second stage is completed, each weight has at most two product terms. Thus, a carry propagation adder can be used to produce the final

result. In the following, we elaborate the details of these two stages.

A. The First Stage

For each lower significance weight, we use a simple OR tree based approximation for power saving. Suppose that the number of inputs is n. If $n \leq 2$, no action is performed. On the other hand, if $n > 2$, we use an OR tree for n-1 inputs to approximate the accumulation result of these n-1 inputs. Thus, after the first stage is done, each lower significance weight has at most two product terms. For each middle significance weight, we use our approximate n:2 compressor for power saving, where n is the number of product terms in this weight. As described in Section II, the designers can choose one of the following two implementations: one implementation is with accurate Sum and approximate Carry and the other implementation is with approximate Sum and approximate Carry. After the first stage is done, each middle significance weight has at most two product terms.

To achieve high accuracy, for each higher significance weight, we use accurate (i.e., exact) 4:2 compressors. For each accurate 4:2 compressor, if the number of product terms is less than 4, the values of other inputs to this compressor are set to be 0. In the rightmost higher significance weight, the carry bit Cin of one accurate 4:2 compressor is from the Carry output of the leftmost middle significance weight, and the carry bit Cin of the other one accurate 4:2 compressor is set to be 0.

## IV. Results

Result of the proposed design is implemented using Xilinx ISE for simulation and Synthesis.

Simulation.

Fig. 6 Simulation.

Synthesis Result:

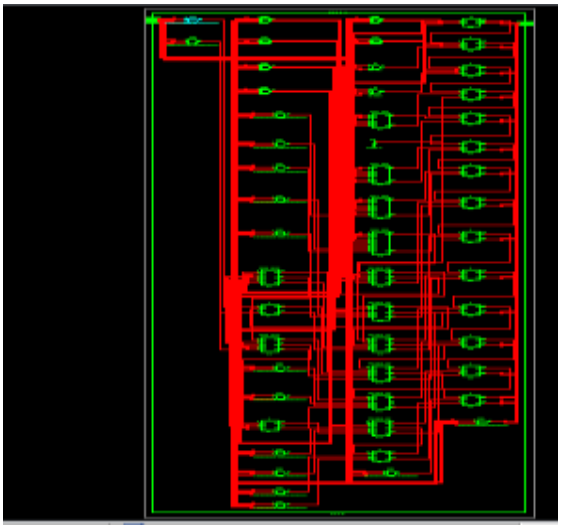


Fig 7 RTL Schematic

| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| Logic Utilization | Used | Available | Utilization |
| Number of Slices | 74 | 4656 | 1% |
| Number of 4 input LUTs | 131 | 9312 | 1% |
| Number of bonded IOBs | 32 | 232 | 13% |

Fig. 8 Design Summary.

```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 18.604ns
```

Fig. 9 Timing Summary.

## V. CONCLUSION

This paper presents a low-power high-accuracy approximate 8 x 8 multiplier design. To achieve high accuracy, we use accurate (i.e., exact) 4:2 compressors in the higher significance weights. To reduce power consumption, we use high-order approximate compressors in the middle significance weights. To our knowledge, the proposed design is the first work that successfully utilizes high-order approximate compressors in the approximate multiplier design for achieving low power dissipation while still maintaining high accuracy.

## REFERENCES

[1] Z. Yang, J. Han, and F. Lombardi, "Approximate Compressor for Error-Resilient Multiplier Design", Proc. of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, 2015.

[2] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and Analysis of Approximate Compressors for Multiplication", IEEE Trans. on Computers, vol. 64, no. 4, pp. 984-994, 2015.

[3] C. Liu, J. Han, and F. Lombardi, "A Low-Power, High-Performance Approximate Multiplier with Configurable Partial Error Recovery", Proc. of IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), 2014.

[4] G. Zervakis, et al., "Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol.24, no.10, pp. 3105- 3117, 2016.

[5] T. Yang, T. Ukezono, and T. Sato "A Low-Power High-Speed Accuracy-Controllable Approximate Multiplier Design", Proc. of IEEE Asia and South Pacific Design Automation Conference (ASPDAC), 2018.

[6] A. Cilardo, et al., "High-Speed Speculative Multipliers Based on Speculative Carry-Save Tree", IEEE Trans. on Circuits and Systems - I, vol. 61, no. 12, pp. 3426–3435, 2014.

[7] J. Liang, et al., "New Metrics for The Reliability of Approximate and Probabilistic Adders", IEEE Trans. on Computers, vol. 62, no. 9, pp. 1760-1771, 2013.

[8] Pakkiraiah cakali, Madhu Kumar Patnala "Design of high speed Brent - Kung based carry select adder" IJSCE March 2013

[9] David h,k hoe, Chris Martinez and sri Jyothsna vundavalli "Design and characterization of parallel prefix adders using FPGAs", Pages.168- 172, march2011 IEEE.

[10] K. Vitoroulis and A.J. Al-Khalili, "performance of parallel prefix adders implemented with FPGA technology," IEEE Northeast Workshop on circuits and systems, pp.498-501, Aug. 2007.

[11] Haridimos t.vergos, Member, IEEE and Giorgos Dimitrakopoulos, Member, IEEE," On modulo 2n+1 adder design" IEEE Trans on computers, vol.61, no.2, feb 2012

[12] Giorgos Dimitrakopoulos and Dimitris Nikolos, "High-Speed Parallel-Prefix VLSI Ling Adders" IEEE Trans on computers, vol.54, no.2, Feb. 2005.