ISSN: 2278-4632
Vol-11 Issue-01 2021

IMPLEMENTATION OF LOW POWER APPROXIMATE MULTIPLIER USING APPROXIMATE HIGH ORDER COMPRESSORS

P. Kumari, P.Anooha, D. Vandana, D.Swathi, P.Ashok Kumar Department of Electronics and Communication Engineering VIGNAN'S INSTITUTE OF ENGINEERING FOR WOMEN

Abstract

To reduce the power consumption, the design of approximate multiplier appears as a promising solution for many error-resilient applications.In this paper, we propose a low-power high-accuracy approximate 8 x 8 multiplier design. The proposed design has two main features. First, according to the significance, different weights utilize different compressors (in different levels of accuracy) to accumulate their product terms. As a result, the power consumption can be saved with a small error. Second, for the middle significance weights, we use high-order approximate compressors (e.g., 8:2 compressor) to reduce the logic of carry chains.To our knowledge, the proposed design is the first work that successfully uses highorder approximate compressors in the approximate multiplier design. Compared with an exact multiplier (Dadda tree multiplier), experimental results show that the proposed approximate multiplier can achieve both low power and high accuracy.

1.INTRODUCTION

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.The common multiplication method is the "add and shift" algorithm. In parallel multipliers, the number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Modified Booth algorithm is one of the most popular algorithms. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages. Further by combining both Modified Booth algorithm and Wallace Tree technique we can see the advantage of both algorithms in one multiplierHowever with increasing parallelism, the amount of shifts between the partial products and intermediate sums to be added will increase which may result in reduced speed, increase in silicon area due to irregularity of structure and also increased power consumption due to increase in interconnect resulting from complex routing.On the other hand "serial-parallel" multipliers compromise speed to achieve better performance for area and power consumption. The selection of a parallel or serial multiplier actually depends on the nature of application. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics.metrics

2.LITERATURE SURVEY

This paper deals with the analysis and design two new approximate 4:2 compressors for utilization in a mutliplier. These designs rely on different features of compression, such that imprecision in computation can meet with respect to circuit-based figures of merit of a design.Different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier.The results show that the proposed designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design. Multipliers are key arithmetic circuits in many such applications such as digital signal processing (DSP).

Approximate computing has received significant attention as a promising strategy to decrease power consumption of inherently error tolerant applications.Here we focus on hardware-leveapproximation by introducing the partial product perforation technique for designing approximate multiplication circuits.The partial product tree of the multiplier is approximated by the proposed tree compressor. In this paper, an approximate multiply-and-accumulate (MAC) unit is introduced. The MAC partial product terms are compressed by using simple ORgates as approximate counters; moreover, to further save energy, selected columns of the partial product terms are not formed. A compensation term is introduced in the proposed MAC, to reduce the overall approximation error.

Approximate/inexact computing has become an attractive approach for designing high performance and low power arithmetic circuits. Floating-point (FLP) arithmetic is required in many applications, such as digital signal processing, image processing and machine learning. Approximate FLP multipliers with variable accuracy are proposed in this paper; the accuracy and the circuit requirements of these designs are analyzed and assessed according to different metrics. It is shown that the proposed approximate FLP multiplier designs further reduce delay, area, power consumption and power-delay product (PDP) while incurring about half ofthe normalized mean error distance (NMED) compared with the previous designs.

3.DADDA MULTIPLIER

3.1 Introduction to dadda multiplier

Multipliers are critical in the present advanced flag handling and for different applications. Numerous scientists have attempted and many are endeavoring to plan the multipliers which will enhance the outline parameters like – speed, low power utilization less range or mix of these in one multiplier by making them appropriate for various fast, low power VLSI usage. The basic idea of DADDA multiplier depends on the underneath framework shape appeared in Fig 2.

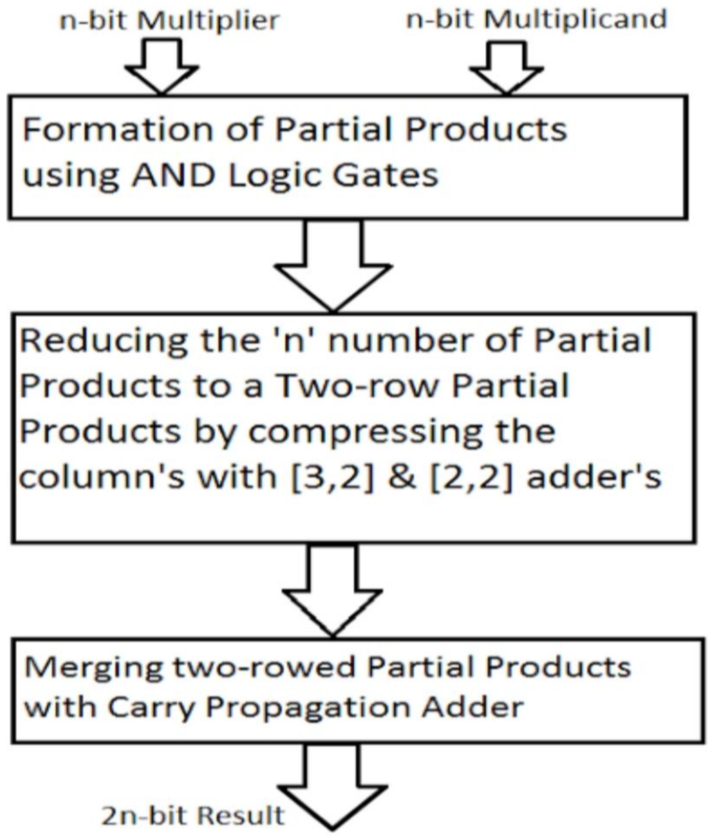

n-bit Multiplier
n-bit Multiplicand
Formation of Partial Products using AND Logic Gates
Reducing the 'n' number of Partial Products to a Two-row Partial Products by compressing the column's with [3,2] & [2,2] adder's
Merging two-rowed Partial Products with Carry Propagation Adder
2n-bit Result

Fig. 1: Flow Chart Of Dadda Multiplier

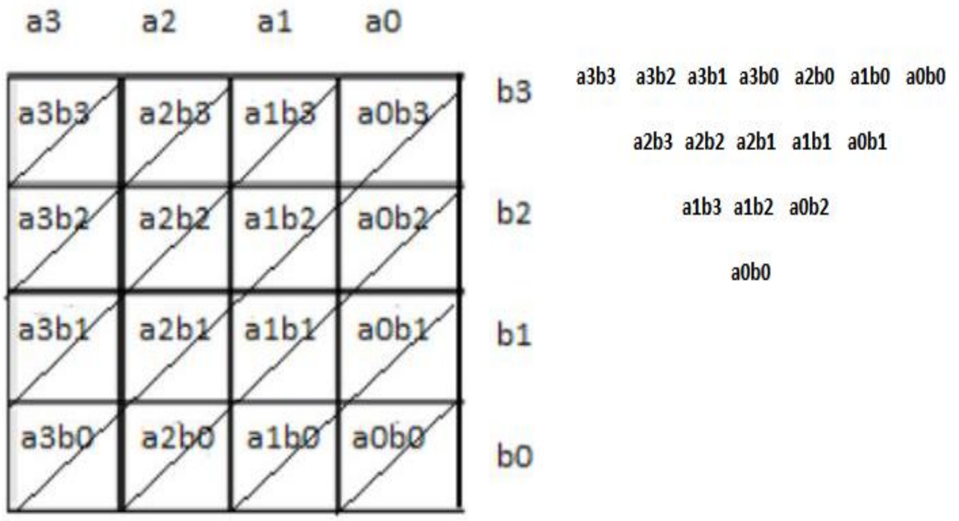


Fig. 2: Architecture Of Dadda Multiplier

3.2 Algorithm Of Dadda Multiplier

Let, us assume the final two-rowed matrix height d1 = 2, based on d1 the successive matrix heights are obtained from dj+1 = 1.5 * dj , where j = 1,2,3,4,............, Rounding of fraction in this matrix height should be done down toleast. i.e, 13.5 = 13(rounded). The matrix heights will be in this fashion 2,3,4,6,9,13,19,28............. Finally the largest dj should be obtained such that derived matrix height shouldn't exceed the Matrix overall height.

1. In the first reduction stage, the column compression is to carried with the [3,2] and [2,2] counters such that the obtained reduced matrix height should not exceed dj.

2. During the compression, the sum is to be passed to same column in the next reduction stage and the carry is to be passed to the next column.

3. The above two steps are to be repeated until a final two-rowed reduced matrix is obtained.

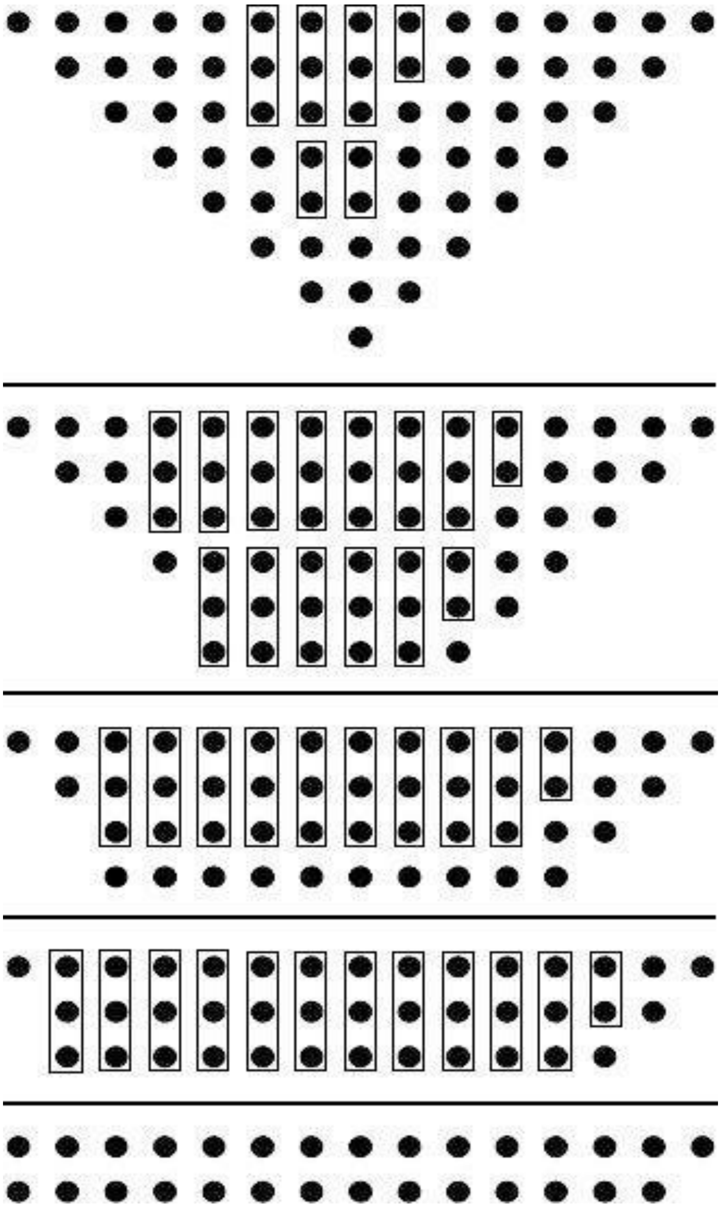


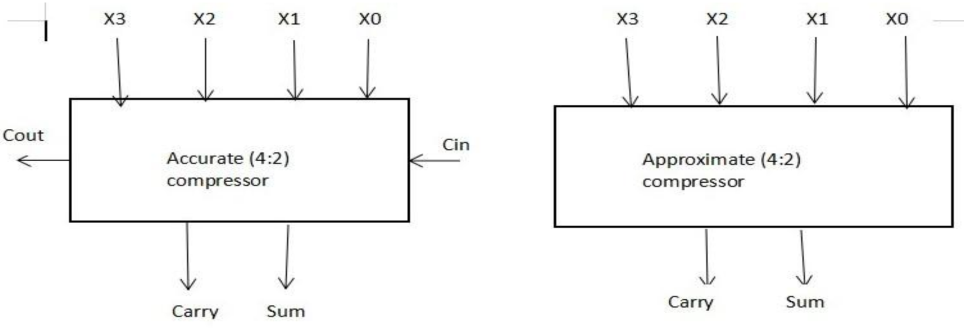
Fig. 3: Algorithm Of Dadda Multiplier

4.Approximate Multiplier Using High order Compressor

4.1 High Order Compressor

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.The 4:2 compressor has 4 inputs X1,X2,X3 and X4 and 2 outputs Sum and

Carry along with a Carry-in(Cin) and a Carry-out(Cout).The input Cin is the output from the previous lower significant compressor.The Cout is the output to the compressor in the next significant stage.



(a) (b)

Fig. 4: (a) Accurate 4:2 Compressor (b) Approximate 4:2 Compressor

4.1.1 Approximation of carry

Half adder, the carry bit Ch is defined as

Ch (X0,X1) = X0. X1

Full adder, the carry bit Cf is defined as

Cf (X0,X1,X2) = X0. X1 + X1. X2 + X0. X2

The Carry output of our approximate 5:2 compressor is

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

The Carry output of our approximate 8:2 compressor is as

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

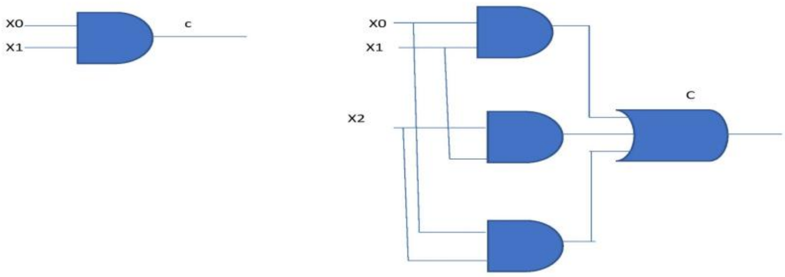


Fig. 5: Modified Halfadder And Modified Fulladder

4.1.2 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. We use the logic gates in SAED 32nm cell library as an example.We find that XOR gate has the largest power, the largest area, and the largest delay. Thus, if we can replace XOR gates with other logic gates, all the design overheads (including the power, the area, and the delay) can be reduced.

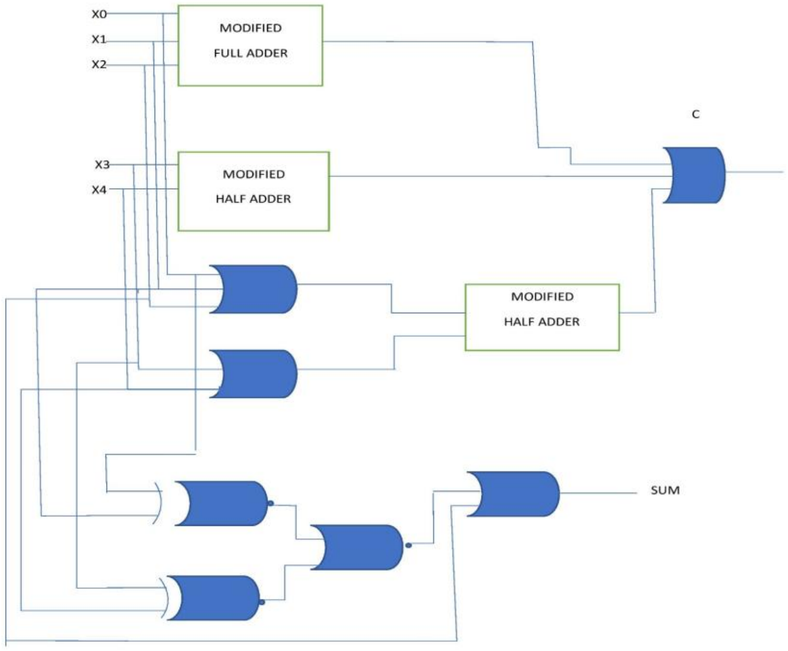


Fig. 6: Sum And Carry Output For 5:2 Compressor

4.2 Approximate Multiplier Design

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 approximate high-order compressors; the lower significance weights use inaccurate compressors (OR-tree based approximation). 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.

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.

The second stage is only for the higher significance weights. In order to achieve high accuracy, we use accurate (i.e., exact) 4:2 compressors to reduce the maximum height of the PPM. The carry bit Cin of the rightmost accurate 4:2 compressor is set to be 0. As shown in Fig. 5, after the second stage is completed, each higher significance weight has two product terms.

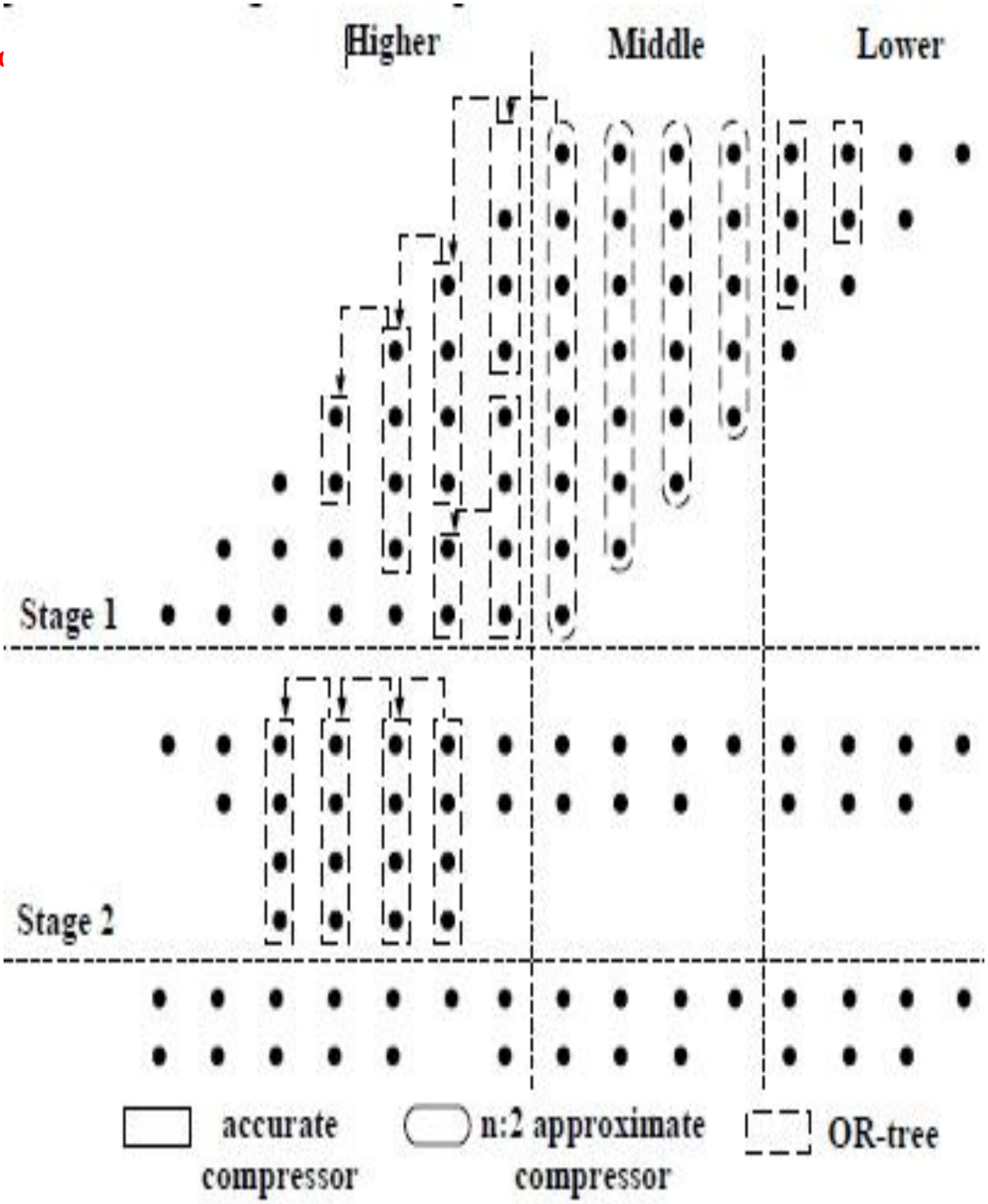
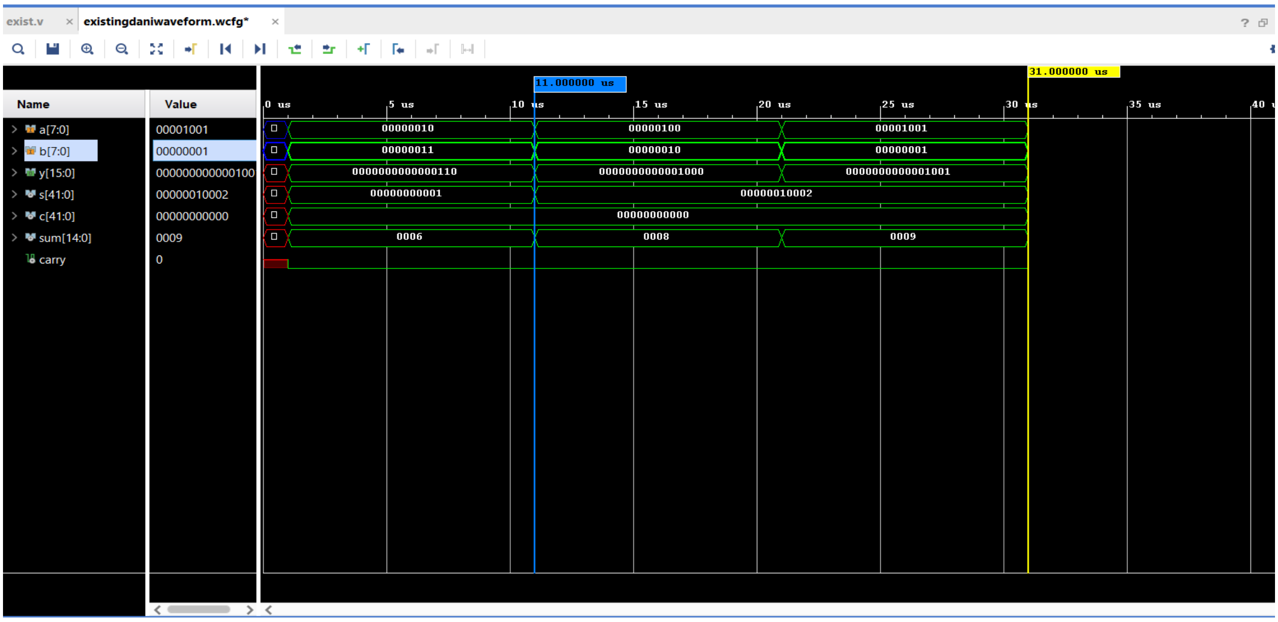


Fig. 7: PPM Reduction In Approximate Multiplier

5.IMPLEMENTATION

5.1 DADDA MULTIPLIER



Name	Value
a[7:0]	00001001
b[7:0]	00000001
y[15:0]	000000000000100
s[41:0]	00000010002
c[41:0]	00000000000
sum[14:0]	0009
carry	0

Fig8: Simulated Waveforms For Dadda Multiplier

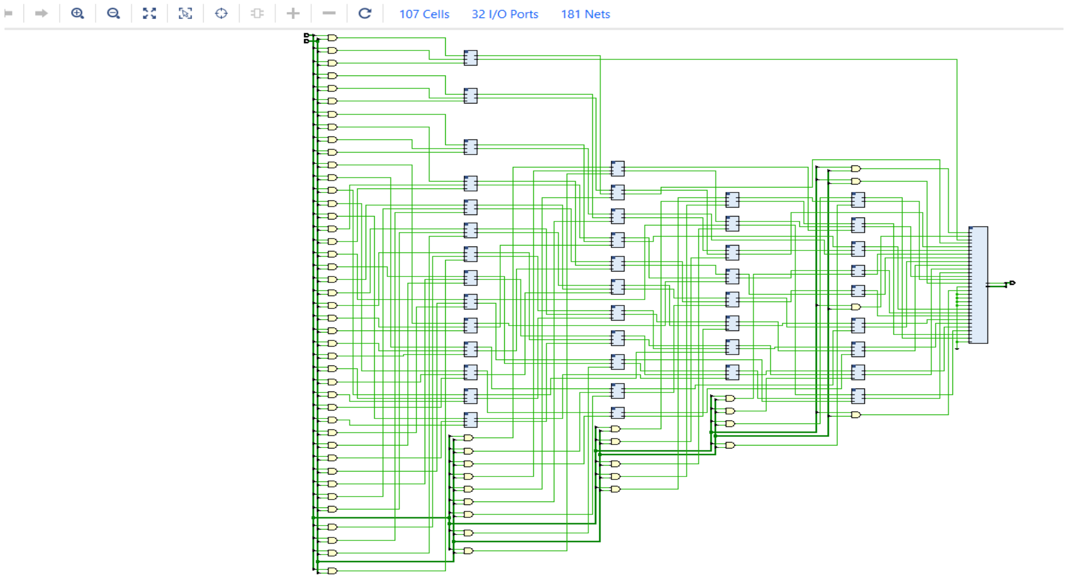
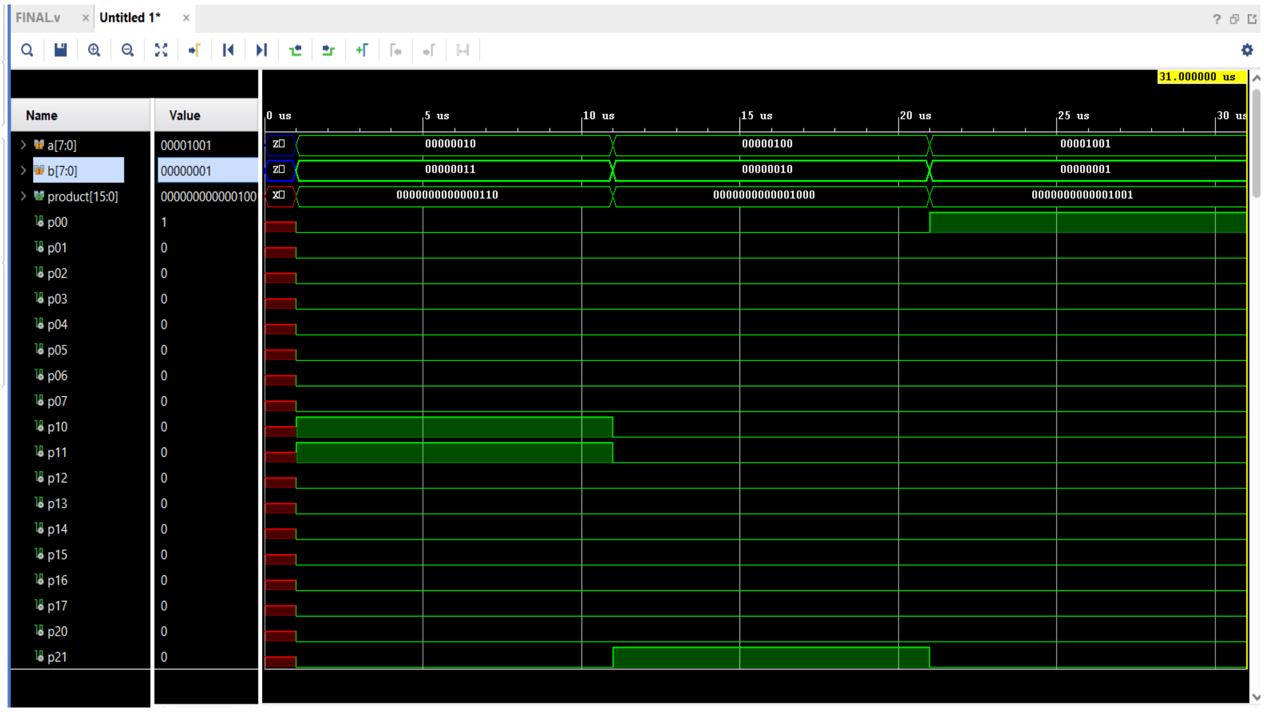


Fig. 9: View Technology Schematic For Dadda Multiplier

5.2 Approximate Multiplier



Name	Value
a[7:0]	00001001
b[7:0]	00000001
product[15:0]	000000000000100
p00	1
p01	0
p02	0
p03	0
p04	0
p05	0
p06	0
p07	0
p10	0
p11	0
p12	0
p13	0
p14	0
p15	0
p16	0
p17	0
p20	0
p21	0

Fig. 10: Simulated Waveforms For Approximate Multiplier

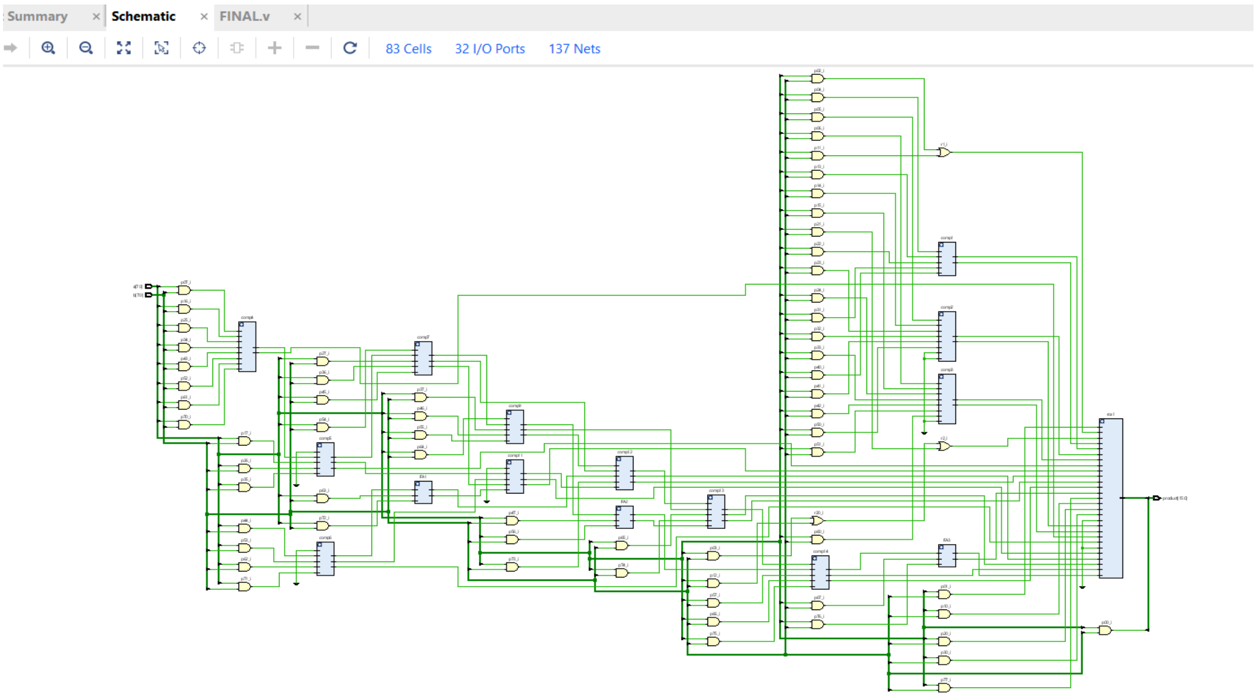


Fig.11: View Technology Schematic For Approximate Multiplier

6. CONCLUSION:

This project 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. The experimental results show that the proposed approximate multiplier design can save area power consumption and high speed compared to normal dadda multiplier. 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

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", IEEETrans. on Computers, vol. 64, no. 4, pp. 984-994, 2015.

[3] C. Liu, J. Han, and F. Lombardi, "A Low-Power, High-PerformanceApproximate 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 AccuracyControllable Approximate Multiplier Design", Proc.OfIEEE 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 andProbabilistic Adders", IEEE Trans. on Computers, vol. 62, no. 9, pp.1760-1771, 2013.

[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.