IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Design and Analysis of Approximate Vedic Multiplier using 3-1-1-2 Compressor

[1] P ANJALI, [2] B SUBBA RAO

[1]Student, [2] Asst. Professor

[1]Electronics and Communication Engineering,

[1]MLEC, Singarayakonda, India

Abstract: In several applications like Digital Signal Processing (DSP), machine learning, image processing applications, the multiplier is the main building block, which needs the more complex design. As the DSP, machine learning and image processing applications are tolerable to outputs with some error; approximate multiplier gives the higher energy and area efficiency as compared to exact multiplier. In this paper a new approximate 3-1-1-2 compressor model is proposed for less complex multiplication process. The proposed compressor design uses less hardware resources and less energy as compared to existing compressors. The simulation and synthesis report shows that the approximate multiplier designed using the proposed approximate 3-1-1-2 compressor consumes less energy and area. When compared with other existing multipliers with bit size of 4×4. The proposed multiplier gives approximately 20% reduction in area and Delay .When compared with the existing multiplier

Index Terms **– Area, Approximate computing, Compressor, Full adder, Machine learning, Multiplier, power.**

I. INTRODUCTION

Now a days as the digital units are becoming hand held and portable there is a great need for systems that consume less power and consume less area utilization but perform the required functionality. By minimizing the power consumption and area utilization of the units of the system we can minimize the overall power consumption and area utilization of the digital system thereby manufacturing it portable. Adders and multipliers are the key components of a DSP blocks. Digital Signal Processing applications are almost in all digital technique nowadays. Applications like filtering, image processing machine learning which process large number of samples to produce the result do not require exact outputs. Due to the restraint of human eye perception the outputs of these applications are tolerable even when it is not exact. These error acceptable DSP applications are used in almost every digital module nowadays for extracting needful information from signals or images. This flexibility in implementation motivated designers to alter Approximate Arithmetic as a workable solution for the design constraint in power consumption and area utilization [1].

In several applications like multimedia, signal processing, machine learning, neural networks and data mining which can not require exact output values (tolerate error), so ,exact computing models are not always require. These units can be replaced with their approximate models. Research on approximate computing for error tolerant applications like DSP, image processing is on the rise. Adders and multipliers form the major components in these applications. Approximate full adders are proposed In [2] at transistor level. They are used in digital signal processing applications and proposed full adder models are utilized in accumulation of partial products in multipliers. To minimize hardware complexity of multipliers, truncation is widely used in fixed-width multiplier models. Then a constant or variable is added output as a correction term to compensate for the quantization error introduced the truncated part in [3], [4].Approximation techniques in multipliers are mainly focus on accumulation of partial products, which is important step in terms of power consumption. in [5] Broken array multiplier is designed , where the LSBs of inputs are truncated, for generating partial products to reduce hardware complexity. The proposed multiplier in [5] saves few adder components in partial product accumulation.

Approximate arithmetic techniques are applied at circuit level or algorithm level to minimize the power dissipation and area utilization of the system. Arithmetic design is by replacing the traditional circuitry with an approximate. The common method of approximate one. Adders and multipliers are the common blocks in any digital system and also power consuming components. Studies have been accompanied to minimize the power and area of the arithmetic circuit by using approximation. In [6] approximate full adders are implemented at transistor level and are used in DSP applications. The proposed adders are used in multiplier units for accumulation. In [7] and [8] truncation is used to minimize the hardware resources utilization and complexity of the multiplier, then a correction term is used to get the final output. Mahdiani et al [9] implemented a lower part OR based adder. They used OR gates to design the approximate arithmetic circuit. A specific bit is selected as critical bit and from that lower part and upper part are selected. The upper part is implemented using full adders and half adders and the lower part is implemented with OR gates to minimize resources. They synthesized using Synopsys software with 0.13µ CMOS technology and noticed a reduction in area and delay as 25.17% and 20.07% compared with the exact multiplier. Qiqieh et al [10] implemented a multiplier using OR gates to collect all the partial products. This method removes all the full adder and half adder blocks used for product collect. They synthesized 128-bit multiplier using 90nm CMOS technology in Synopsys design compiler and obtained a 65% reduction in delay and 45% reduction in area consumption compared to 128-bit exact multiplier. Liu et al [11] designed a

multiplier using approximate adders which processes data in parallel and also designed an error recovery circuit to minimize the accuracy loss due to the usage of approximate adders. They synthesized the multiplier using 28nm CMOS library and observed a reduction of 20% in critical path delay and up to 69% in power dissipation when compared with the exact Wallace tree multiplier. Inexact adders and approximate 4-2 compressors are used S.Venkatachalam in [12] et al to collect the partial products. The original partial product matrix is changed and all the columns are collected using approximate adders. The implimented 16-bit approximate multiplier design is synthesized using TSMC 65 nm library in Synopsys tool and noticed a minimization in power of 38% when compared with the general multiplier. an approximate 2x2 multiplier is proposed by Kulkarni et al in [13] and used it to design a larger multiplier. A 2x2 multiplier was implemented using approximate half and approximate full adders. Larger multipliers were built using the 2x2 multiplier. The implemented approximate 16-bit multiplier design is synthesize using Cadence RTL compiler and the achieved a minimization of 31.8% in power dissipation when compared with the conventional multiplier. The rest of this paper is organized as follows. Section II conveys a review of the 3-1-1-2 compressor. in Section III conveys The proposed designs. Then the simulation results and synthesis report of proposed compressors and multipliers are presented and analyzed in Section IV. And finally, the conclusion is presented in Section V.

II. EXISTING 4-2 COMPRESSOR

The compressors are used to design the high speed multipliers, to minimize the delay in the summation part of the partial product. Therefore, the compressors improve the multiplier circuits and overall systems performance . The 4-2 compressor is implemented using XOR-XNOR gates and 2:1 multiplexers. This 4-2 compressor has five inputs in which one carry input and two outputs. The 3-2 compressor has three inputs and two outputs; it is similar to FA(full adder).

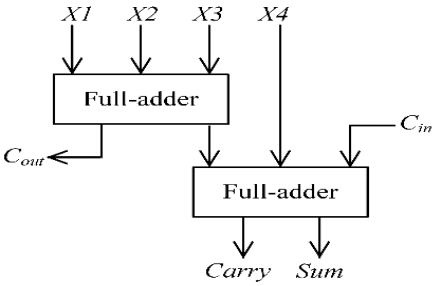


Fig.1. conventional existing 4-2 compressor

SUM=X1^X2^X3^X4^Cin; (1)

CARRY=((X1^X2^X3^X4)&Cin) | ((~(X1^X2^X3^X4))&X4) (2)

Cout=((X1^X2)&X3) | ((~(X1^X2))&X1)) (3)

Table 1: Truth table for conventional existing 4-2 compressor

A_1	A_2	A_3	A_4	C_{IN}	C_{OUT}	$CARRY$	SUM
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	0	1	0
0	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	1
0	1	1	1	0	1	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	0
1	0	0	1	1	0	1	1
1	0	1	0	0	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	0	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	0	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	0	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

III. EXISTING 3-1-1-2 COMPRESSOR

The 3-2 compressor contains three inputs and two outputs; it is similar to the full adder. The 4-2 compressor contain five inputs and two outputs. In the lower part of the multiplier summation , three input and one carry input compressor are require to compute summation result with less delay. To overcome this problem, the 3-1-1-2 compressor is designed using K-map and Boolean expression.

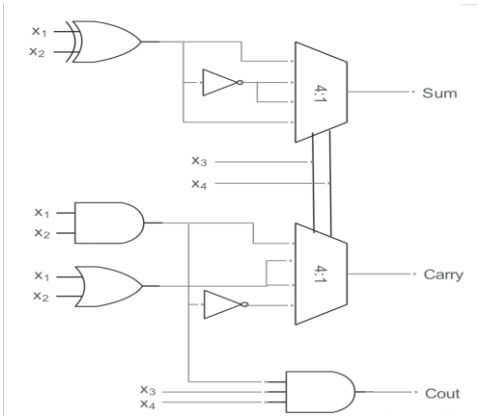


Fig.2. Existing 3-1-1-2 compressor [14]

IV. PROPOSED APPROXIMATE 3-1-1-2 COMPRESSOR

The proposed high speed area-efficient approximate 3-1-1-2 compressor is implemented in this section. The existing 3-1-1-2 compressor consumes more area and power in order to minimize the resources utilization. Approximation is introduced in the design. The approximate compressor accepts four inputs and produce two outputs are sum and carry. The compressor inputs are X1, X2, X3 and X4, outputs are Carry' and Sum. In the design of approximate 3-1-1–2 compressor Sum and carry can be generated using a multiplexer (MUX) based design approach. Output of XOR gate(X3 XOR X4) acts as the select line for the MUXes. When select line goes high, (X1 XOR X2) is selected and when it goes low, (X1 XNOR X2) is selected for sum operation. For carry operation when select line goes high, (X1 OR X2) is selected and when it goes low, (X1 AND X2) is selected.

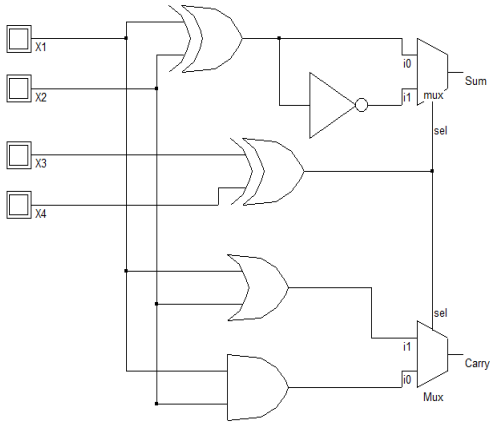


Fig.3. proposed 3-1-1-2 compressor

V. PROPOSED MULTIPLIER

In this section, the proposed design of a 4 × 4 approximate Vedic multiplier is presented. One of the fastest arithmetic processing method is The Vedic multiplication. This is implemented and used by Indian mathematicians. The 4 × 4 multiplier is designed based on the Vedic mathematics using proposed compressor, in first stage bitwise logical AND operation is performed for commutating partial products. In the first stage, sixteen logical AND gates are used from least significant Bit(LSB) to most significant bit(MSB) to compute single bit output. After, half adders, full adders and compressors are used to compute multiplication result from S0 to S7 with carry propagation manner. The main aim of replacing all general compressors with approximate compressors is to minimize the delay, power and area significantly.

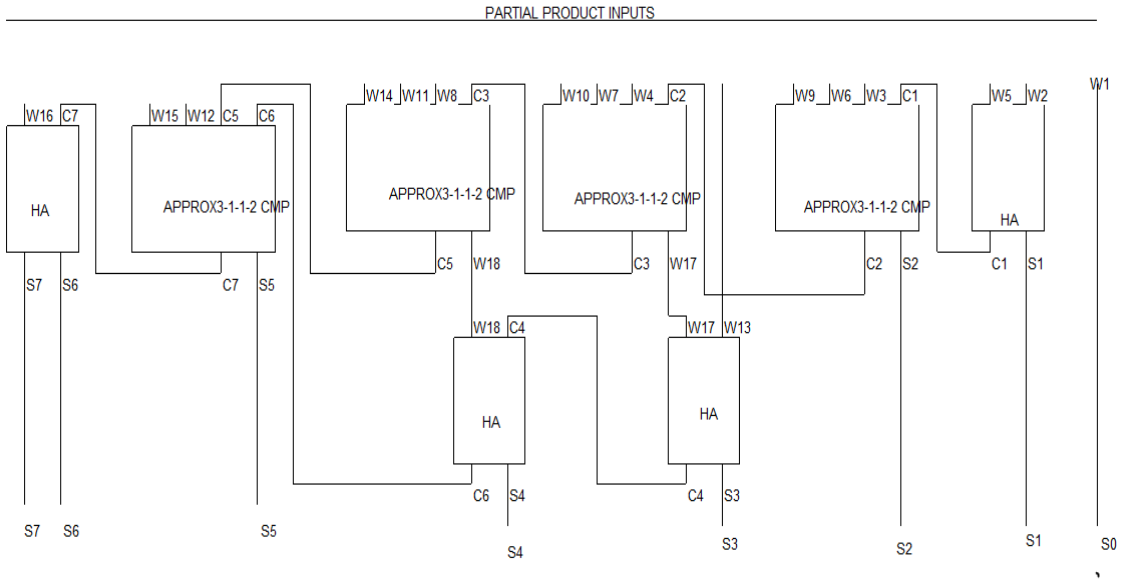


Fig.4. Proposed Vedic Multiplier

VI. SIMULATION RESULTS AND DISCUSSION

In this section proposed approximate compressor and multiplier is simulated and synthesis by Xilinx ISE 14.7 and vivado 2019.2 are discussed. For the better comparison propose between existing Vedic multiplier in [14] and proposed multiplier is implemented using Xilinx ISE 14.7

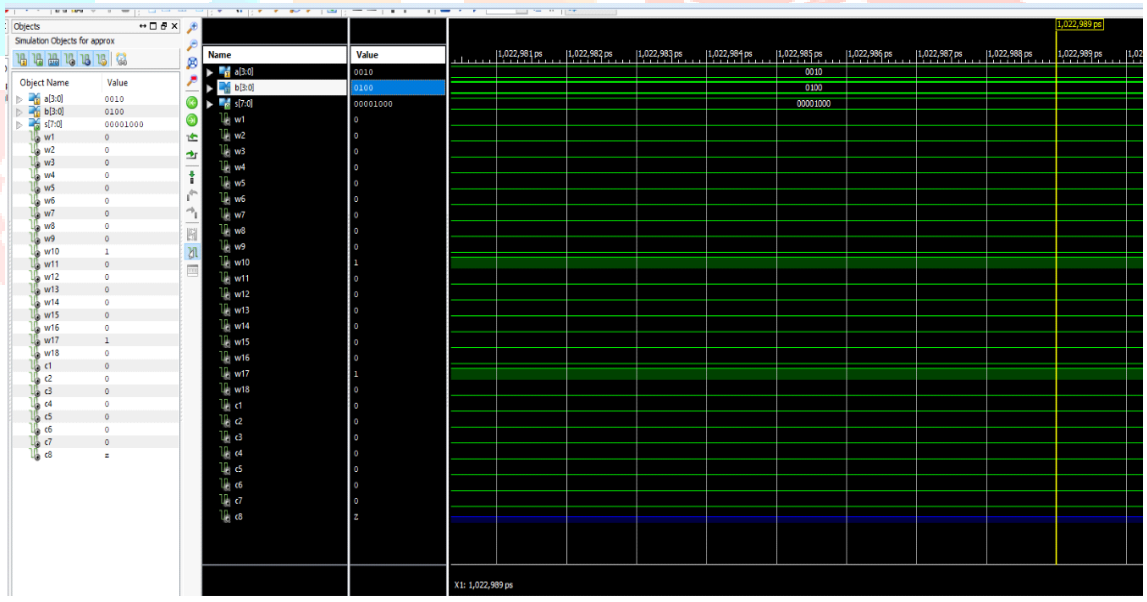
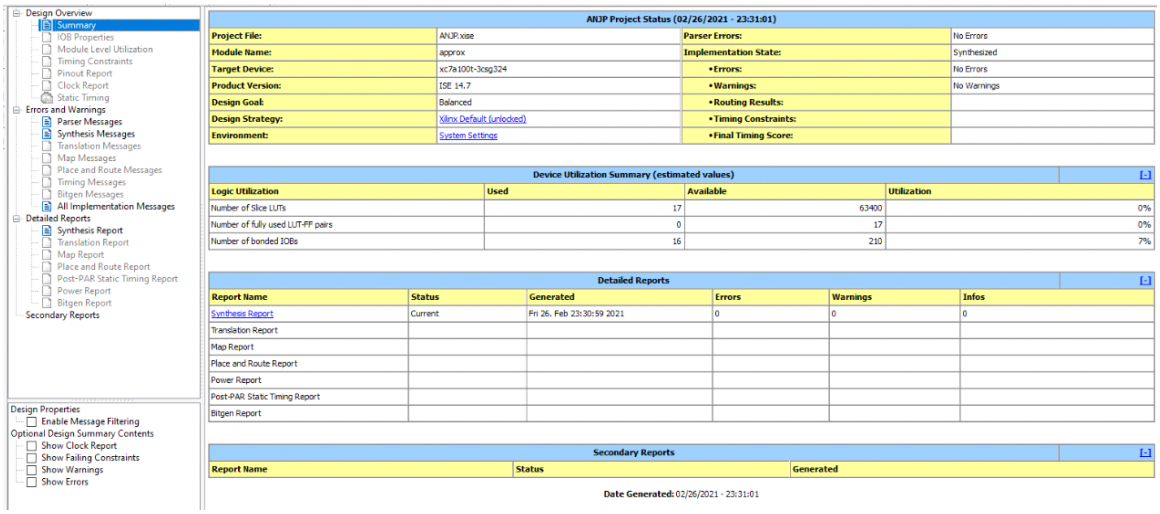


Fig.5. Summation result of proposed approximate multiplier



Design Overview
- Summary
- IOB Properties
- Module Level Utilization
- Timing Constraints
- Pinout Report
- Clock Report
- Static Timing

Errors and Warnings
- Parser Messages
- Synthesis Messages
- Translation Messages
- Map Messages
- Place and Route Messages
- Timing Messages
- Bitgen Messages
- All Implementation Messages

Detailed Reports
- Synthesis Report
- Translation Report
- Map Report
- Place and Route Report
- Post-PAR Static Timing Report
- Power Report
- Bitgen Report

Secondary Reports

Design Properties
- Enable Message Filtering

Optional Design Summary Contents
- Show Clock Report
- Show Failing Constraints
- Show Warnings
- Show Errors

ANJP Project Status (02/26/2021 - 23:31:01)

Project File:	ANJP.xise	**Parser Errors:**	No Errors
Module Name:	approx	**Implementation State:**	Synthesized
Target Device:	xc7a100t-3csg324	**•Errors:**	No Errors
Product Version:	ISE 14.7	**•Warnings:**	No Warnings
Design Goal:	Balanced	**•Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	
Environment:	System Settings	**•Final Timing Score:**	

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	17	63400	0%
Number of fully used LUT-FF pairs	0	17	0%
Number of bonded IOBs	16	210	7%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Fri 26. Feb 23:30:59 2021	0	0	0
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports

Report Name	Status	Generated

Date Generated: 02/26/2021 - 23:31:01

Fig.6. Area report of proposed approximate multiplier

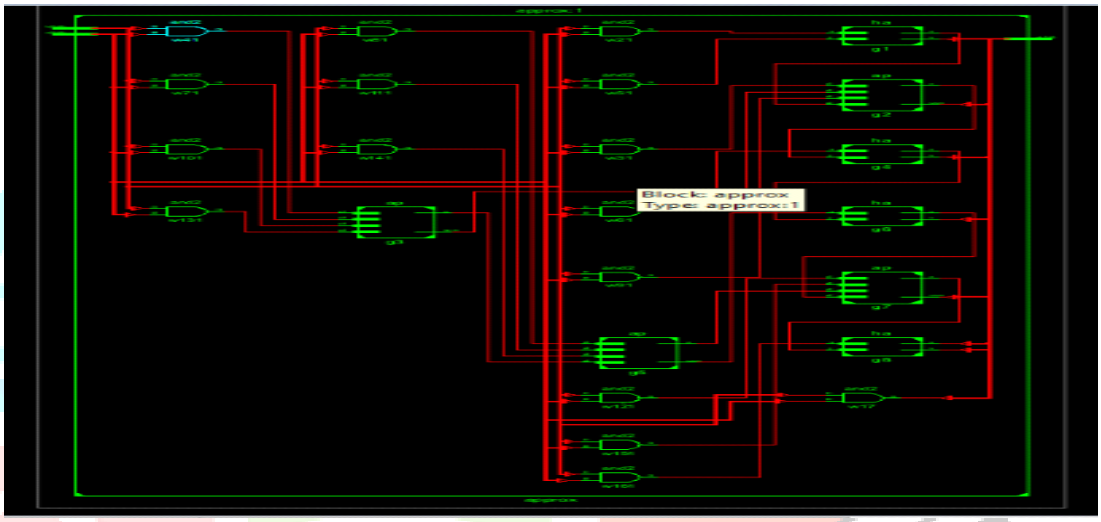


Fig.7. RTL view of proposed approximate multiplier

```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 144 / 8
-------------------------------------------------------------------------
Delay:               2.620ns (Levels of Logic = 6)
  Source:            a<3> (PAD)
  Destination:       s<7> (PAD)

  Data Path: a<3> to s<7>
                                Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             5    0.001    0.398  a_3_IBUF (a_3_IBUF)
     LUT2:I0->O            2    0.097    0.688  w41 (w4)
     LUT6:I1->O            2    0.097    0.561  g5/Mxor_w3_xo<0>1 (g5/w3)
     LUT6:I2->O            3    0.097    0.305  g7/Mxor_w3_xo<0>1 (g7/w3)
     LUT5:I4->O            1    0.097    0.279  g7/g1/Mmux_d11 (s_5_OBUF)
     OBUF:I->O                  0.000            s_5_OBUF (s<5>)
    ----------------------------------------
    Total                      2.620ns (0.389ns logic, 2.231ns route)
                                       (14.8% logic, 85.2% route)

=========================================================================
```

Fig.8. Delay report of proposed approximate multiplier

From the synthesis report it is observed that Existing Vedic multiplier takes 20 LUTs for the designing the system but Proposed Vedic multiplier is designed using 17 LUTs ,so 15% area efficient than the existing multiplier and Existing Vedic multiplier Delay is 3.501ns for the system but Proposed Vedic multiplier Delay is 2.62ns ,so 25% Delay is reduced in proposed system when compared with existing system. From that it is noticed proposed system is faster than existing system. From the overall report proposed system is optimized in aspect of Area and Delay as compared to existing system.

VII. CONCLUSION

In this paper, we proposed a method to design approximate 3-1-1-2 compressor. by using proposed compressor Vedic multiplier is designed. This approach minimize the active logic gate count compared with the exact 3-1-1-2 compressor, the proposed compressor designs achieve a significant reduction in power consumption, area utilization, and delay. From Simulation results and synthesis report it is indicate that the utilization of the proposed models brings significant improvement to the resource performance of multipliers. At the same time, the proposed multipliers keep a tolerable error performance.

In conclusion, this work has shown that multiplier can be used for approximate computing by an approximate design of a compressor; this proposed multiplier gives better results in terms of design parameters compared to existing multipliers, and in terms of accuracy metrics, critical path delay, area and power.

REFERENCES

[1].J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design", 18th IEEE European Test Symposium (ETS), 2013.

[2].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.

[3] 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.

[4] 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.

[5] 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.

[6].V. Gupta, D. Mohapatra and A. Raghunathan, "Low-Power Digital Signal Processing using Approximate Adders", IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems, vol. 32, no. 1, January 2013.

[7].E. J. King and E. E. Swartzlander, "Data-dependent truncation scheme for parallel multipliers", Conference Record of the Thirty-First Asilomar Conference on Signals Systems and Computers (Cat. No.97CB36136), vol. 2, pp. 1178-1182, 1997.

[8]. Kyung-Ju Cho, Kwang-Chul Lee, Jin-Gyun Chung and K. K. Parhi, "Design of low-error fixed-width modified booth multiplier", IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 12, no. 5, pp. 522-531, May 2004.

[9].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 Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 4, pp. 850-862, April 2010.

[10].I. Qiqieh, R. Shafik, G. Tarawneh, D. Sokolov and A. Yakovlev, "Energy-efficient approximate multiplier design using bit significance-driven logic compression", Design Automation & Test in Europe Conference & Exhibition (DATE) 2017, 2017.

[11].C. Liu, J. Han and F. Lombardi, "A low-power high-performance approximate multiplier with configurable partial error recovery", 2014 Design Automation & Test in Europe Conference & Exhibition (DATE), 2014.

[12].S. Venkatachalam and S. Ko, "Design of Power and Area Efficient Approximate Multipliers", IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 25, no. 5, May 2017.

[13].P. Kulkarni, P. Gupta and M. Ercegovac, "Trading Accuracy for Power wth an Underdesigned Multiplier Architecture", 2011 24th Internatioal Conference on VLSI Design Chennai, pp. 346-351, 2011

[14]. K. Sivanandam , P. Kumar," Design and performance analysis of reconfigurable modified Vedic multiplier with 3-1-1-2 compressor",Elsevier Microprocessors and Microsystems,pp.97-106,2019.