Design and Implementation of Sixteen-bit Low Power and Area Efficient Dadda Multiplier

Manu V
M.Tech (VLSI), Department of ECE
Bangalore Institute of Technology
Bangalore, Karnataka, India
manuvvasu@gmail.com

Vijaya Prakash A M
Professor, Department of ECE
Bangalore Institute of Technology
Bangalore, Karnataka, India
am_vprakash@yahoo.co.in

Chandra Mohan U
Chief Technology officer
Banashree RENEWABLE ENERGY System PVT. Ltd.
Bangalore, Karnataka, India
cumapathy@banashreesystems.com

***Abstract*— Low power and area efficient 16-bit multiplier has been designed and implemented using the Dadda algorithm. Here, the prime building block having low power dissipation and area efficient optimized full adder architecture and Carry Look-Ahead (CLA) adder is designed and implemented. Designing full adder is done by making use of complex cells in the technology node of 65nm to reduce the power dissipation and minimum area using TSMC 65nm library. The proposed multiplier design is optimized, simulated using ISE simulator and synthesized using Cadence Genus EDA tool and results are demonstrated. The power and area of the Dadda multiplier designed using proposed full adder is minimum compared to conventional design. The power and area are improved by an amount of 15.32% and 1.91% respectively, than the conventional-full adder. Dadda Multiplier designed here is used to implement 16-bit ALU the power and area obtained are 20.65% and 1.8% lesser than the existing design.**

Keywords—Dadda Multiplier, ALU, Carry Look-Ahead Adder (CLA), Power, Area.

I. INTRODUCTION

Now a days, usage of portable electronic devices area increasing each day, portable devices require battery for operating them. So power dissipation becomes the important concern that has to be taken care in designing of such devices i.e., laptops, mobile phones, tablets, notebooks, and many more personal electronic devices. The power dissipation plays a fundamental role in VLSI technology. [8]More power dissipation leads to more heating of the circuits resulting in decreasing battery life span and also need to provide cooling for the circuit. Therefore, power dissipation affects the battery life span and overall system cost increases. Most of all digital electronic devices discussed are used in the applications like DSP's, microcontrollers, video processing and image processing. Various arithmetic and logic operations are used to perform addition, multiplication, subtraction, division, shifting, rotate, etc. The sever necessity for low power dissipation had been a high issue in all embedded processor designs. Power reduction for any system or designs can be satisfied at different design levels, such as threshold voltage scaling and transistor sizing at the semi-conductor chip design stage, power gating and clock gating at the logic level and register transfer level, Dynamic scaling of voltage at the system level. Reduction of power can be achieved on particular functional part or components of any processor.

The design of high speed and low power microprocessors require sub-parts that consumes lesser power. Arithmetic and Logic Unit (ALU) is one part of the microprocessor that consumes majority of the power. Therefore, in order to reduce the power dissipation of the ALU, each of its subparts must consume lesser power. An ALU carryout arithmetic operations such as subtraction, addition, division, multiplication, etc., and logical operations such as Ex-OR, NAND, AND, OR, Logical shift, Rotate, etc.

Multiplication is the one of the common and prime operation used in most of the all-electronic applications and in numerous digital communication applications. [6]Multipliers with lesser power consumption, area and low latency are always used to design an optimized digital circuit in such a way that, in minimum time response maximum throughput can be obtained. Full adders and half adders are the prime building blocks of any multiplier design. Till now different design architectures of half adder and full adder are designed and implemented to decrease the power dissipation, area and delay in order to get an efficient multiplier circuit. Along with this, different algorithms have been introduced to achieve optimized power, area and delay such as Dadda algorithm, Wallace tree, Booth multiplier and Vedic algorithms. [3]The multipliers used recently are Dadda Algorithm and Reduced-sp-D3L$_{sum}$ (reduced-split pre-charge data driven dynamic sum logic) adder logic technique. These designs has lesser power dissipation while operating at higher frequency comparatively, but still, power dissipation has to be reduced, so it will reduce overall power dissipation in the bigger circuits where multiplier becomes the prime building block.

II. LITERATURE SURVEY

Study on reduction of power dissipation in the digital system design has being carried out till now. There are two types of power dissipation in digital systems with CMOS technology. Switching activity power dissipation called dynamic power dissipation and leakage power dissipation due to leakage current called static power dissipation in the transistor. [16]Different techniques have been applied to reduce

Work carried out in Banashree RENEWABLE ENERGY System PVT. Ltd and Bangalore Institute of Technology

978-1-7281-0630-4/19/$31.00 ©2019 IEEE

the dynamic power dissipation such as reducing switching frequency, reducing switching capacitance, or reducing supply voltage. Similarly, leakage power reduction is done by applying certain techniques such as reducing supply voltage, by reducing circuit size, by reducing operating temperature or by increasing transistor threshold voltage.

Power dissipation is a critical designing problem in most of the embedded processor design. One of the common and main part is Arithmetic and Logic Unit in the processor. Usually, ALU's are implemented using a combinational logic circuit containing more number of functional components for performing different logic and arithmetic operations. ALU's can be designed with a chain or a tree structure. [16] This can be easily modeled or introduced into a processor design environment, so that overall power dissipation is reduced efficiently for a given application. The results show that the power improvement ranges from 43.5% to 49.6% on average and maximum 46.9% of ALU power reduction can be obtained. Multiplier with optimized full adder designed using pass-transistor logic technique since it uses lesser transistors count and small node capacitances and that introduces minimum delay and speed of operation can be increased.

To increase speed and to minimize the power the Dadda multiplier used with different compressors. The use of compressors in the multipliers not only reduces the vertical critical path but also reduce the stages of operation simultaneously. [8]The speed of the Dadda multiplier is improved by introducing different compressors instead of 4:2 compressors. In this paper 4:3, 5:3, 6:3 and 7:3 compressors are used to for reducing the number of stages of addition in the multiplication algorithm by reducing the number of half adders and full adders. [10]Different architectures of full adders are designed using two 2-input MUX for generating both sum and carry, two 4-input MUX combination for producing sum bit and carry bit, two 2-input XOR gates followed by two 2-input MUX combination. [1]The model of 4bit multiplier having high speed of operation and low power using Dadda Algorithm are designed using pass transistor logic.

III. Proposed Design Methodology

Different technology approaches have been involved in reducing the total power consumption in the VLSI chip design. Design approaches in each and every category can be even further classified depending upon the design stages such as transistor sizing, Logical level, Technology Mapping, power gating and RTL level clock gating. Once the optimized logic equations are obtained, then the equations are mapped on to a library that contains improved logic gates characteristics in the specified technology node. A typical library contains numerous logic gates with different transistor sizes having different drive strengths. Recent technology mapping method uses graph covering formulation to target power area and delay.

In this paper, proposed design implementation is based on the technology mapping technique at logic gate level is often applied for reduction of power. Here, TSMC65nm library is used for designing and implementation. Initially the library was analyzed for the complex gates or cells for its functionality. Depending on the existing or required functionality, the new architecture has been developed for taking into consideration of Power, Area and Time delay. Initially, the architecture for full adder has been developed and verified for the functionality based on the library cells (technology mapping) and minimal power, the architecture is implemented. The 16-bit multiplier is designed using Dadda algorithm. The multiplier is based on the optimized full adder architectures designed by analyzing the technology library and by using complex gates, so as to minimize the power dissipation and area which intern reduces the overall power and area. A 16-bit Arithmetic and Logical Unit is designed and implemented using the proposed Dadda Multiplier having the new adder architecture obtained from technology mapping technique and CLA which shows power dissipation and area comparatively lower.

A. Implementation of Carry Look-Ahead adder (CLA)

The CLA is faster in operation when it is compared to the ripple carry adder. It improves the speed in such a way that, by reducing the amount of time required for determining carry bits. The CLA generates one or more carry bits first and then the sum bit, which reduces the time required for evaluating the result of higher order bits. Here, concept of propagating a carry i.e., generated from previous stage and generating a carry of that particular stage is used. For every bit to be added, CLA logic will evaluate whether to generate the carry or propagate the carry. This lets the circuit to preprocess the two numbers or bits being added to find out the carry ahead of time.

The grouping of the propagate signals and generate signals is done to form the higher order bit carry directly. The generate (gi) and propagate (pi) signals are calculated by

Consider if a_i and b_i are two bits, then carry generation and propagation of carry is given by,

$$gi = (a_i \,\&\, b_i) \quad (1)$$

$$pi = (a_i \,\wedge\, b_i) \quad (2)$$

Next, while performing original addition there is no delay in waiting for the ripple carry effect or time taken for the carry generated from the initial adder to be passed to the last adder.

For a 4-bit adder, carry generation at each stage is given by

$$C0 = g0 + p0Cin \quad (3)$$

$$C1 = g1 + p1C0 = g1 + p1\,(g0 + p0\,Cin) = g1 + p1g0 + p1\,p0\,Cin \quad (4)$$

$$C2 = g2 + p2C1 = g2 + p2g1 + p2\,p1\,g0 + p2\,p1\,p0\,Ci \quad (5)$$

$$C3 = g3 + p3C2 = g3 + p3g2 + p3\,p2\,g1 + p3\,p2\,p1\,g0 +$$

p3 p2 p1 p0 Cin (6)

Fig.1 shows a 16-bit CLA is designed using four 4-bit CLA in a ripple carry manner which intern reduces the power dissipation.

B. Implementation of 16-bit Dadda multiplier

Different multiplier algorithms have been designed in such a way that the overall propagation delay has been reduced for adding partial products. The most effective algorithm is Dadda algorithm. Figure 2 shows, the 16-bit Dadda multiplier stage reduction algorithm and it has 256 partial products, so, the number of rows of the tree is 16. The Dadda algorithm performs stage by stage addition required for the summing of the partial products. Dadda Algorithm is applied in order to reduce the height of the row from 16 to 2 in number of stages so that final stage has the product terms. In first stage of addition 15 full adders and 5 half adders are used and number of row count is reduced to 12 from 16. In second stage, 33 full adders and 3 half adders are used and the row is reduced to 9. In third stage, 51 full adders and 3 half adders are used and row reduced to 6. In fourth stage, 44 full adders and 2 half adders are used then the row reduced from 6 to 3, in fifth stage 25 full adders and a half adder is used and row is reduced to 2 and at the final stage of addition having 2 rows. In case of other algorithms, need to wait for the preceding stage to be executed so that, the succeeding stages use the carry of the earlier stage which will increase the propagation delay. In 16-bit Dadda multiplication the reduction and product obtained in six stages. The resulting bits in a final matrix having of two rows of which can be summed using a multiple bit adder (e.g. carry look ahead adder, carry skip adder, ripple carry adder). It is observed that in each stage of reduction full adders and half adders operate simultaneously, which results in faster execution of multiplier. After all the reduction stages, at the end two rows of bits are available that can be added using multi-bit adders. The dadda multiplier is simulated using ISE simulator for functionality and synthesized using Cadence GENUS EDA tool.

It is observed that in each stage of reduction full adders and half adders operate simultaneously, which results in faster execution of multiplier. After all the reduction stages, at the end two rows of bits are available that can be added using multi-bit adders.

In Dadda algorithm, in common, the total of full adders required is N2 - 4N + 3 and the number of half adders is N - 1, where N represents the number of bits. The number of reduction stages for implementing Dadda architecture varies depending upon the number of bits is shown in Table 1. For a 16-bit multiplier the number of full adders used is 195 numbers and 15 numbers. The above fig.3 shows the RTL schematic representation of the 16-bit Dadda Multiplier using carry look-ahead adder.

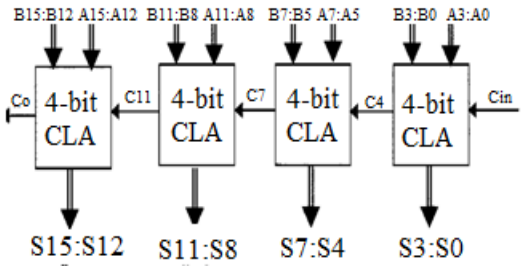

Fig.1. 16-bit Carry Look-Ahead Adder

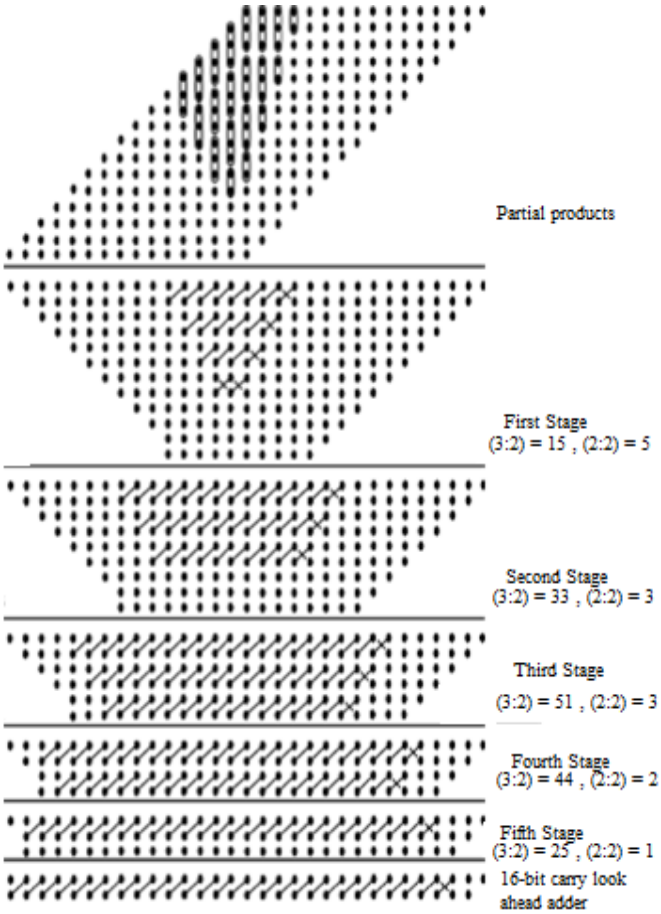

Fig.2. 16-bit Dadda Multiplier stage reduction algorithm

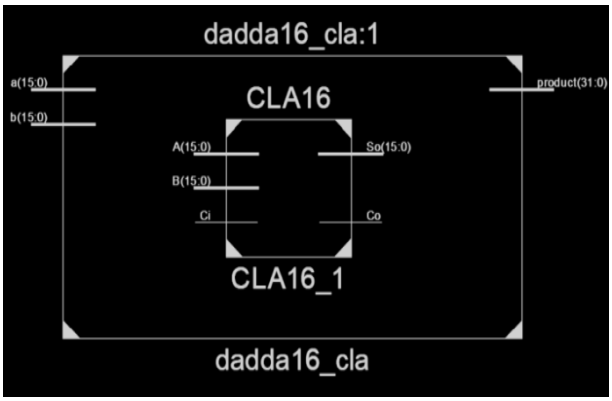

Fig.3 RTL schematic Representation of 16-bit Dadda Multiplier

Table I showing the number of stages of reduction in Dadda multiplier

No. of bits in multiplier	Number of reduction stages
3	1
4	2
$5 \leq N \leq 6$	3
$7 \leq N \leq 9$	4
$10 \leq N \leq 13$	5
$14 \leq N \leq 19$	6
$20 \leq N \leq 28$	7
$29 \leq N \leq 42$	8
$43 \leq N \leq 63$	9

C. Implementation of Arithmetic and Logical unit (ALU) using Dadda multiplier

Arithmetic Logic Unit (ALU) is the main Central Processing unit which carryout arithmetic operations like addition, subtraction, multiplication, division of numbers and logical operation as AND,OR, Logical shift, Rotate and other functions.

In this paper, a 16-bit ALU is designed and implemented that perform ten operations such as Addition, Subtraction, Multiplication, Division, Greater than and equal to as arithmetic operations and logical shift left, logical shift right, rotate right, rotate left as logical operations as shown in fig.4. Table II shows the operations performed by the ALU. Adder/subtraction module is designed using carry look ahead adder (CLA), Multiplier module is designed using Proposed Dadda Multiplier. The 16-bit ALU is simulated using ISE simulator for functionality and synthesized using Cadence GENUS EDA tool.

D. Prime building block of Dadda multiplier

In proposed model Full adder has been modified in such a way that total power consumption is minimized and has the lesser area, by using technology mapping technique, then the full adder is implemented to dadda multiplier which results in lesser power dissipation and minimum area. The schematic of the proposed full adder is shown in Fig.5. The full adder is designed using optimized XOR using complex gate available in library. The functionality of proposed full adder is verified using ISE simulator.

Table II Operations performed by ALU

Select Input vectors (S3 S2 S1 S0)	Operation performed by ALU
0 0 0 0	Addition
0 0 0 1	Subtraction
0 0 1 0	Multiplication
0 0 1 1	Division
0 1 0 0	logical shift left
0 1 0 1	logical shift right
0 1 1 0	rotate left
0 1 1 1	rotate right
1 0 0 0	greater
1 0 0 1	equal

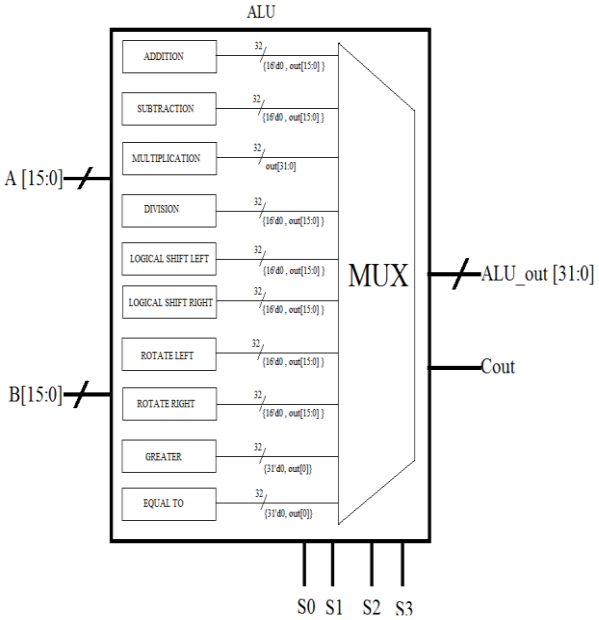

Fig.4 Block diagram Showing Operations performed by ALU

Full adder functionality is

$$S = (A \wedge B \wedge C) \tag{7}$$

$$Co = ((A\&B) \mid (B\&C) \mid (C\&A)) \tag{8}$$

The proposed full adder function is given by,

Functionality for generating sum bit,

$$W1 = \sim ((\sim (A \mid B)) \mid (A \& B)) \tag{9}$$

$$S = \sim ((\sim (W1 \mid C)) \mid (W1 \& C)) \tag{10}$$

The functionality for generating carry bit,

$$Co = ((A \& B) \mid (W1 \& C)) \tag{11}$$

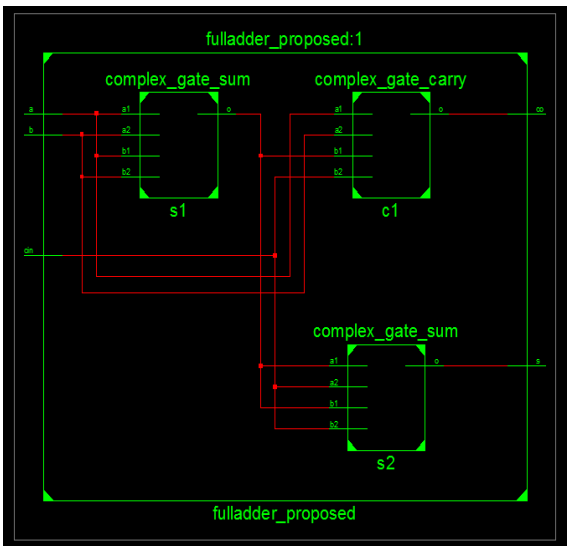

Fig.5 RTL schematic of proposed full adder

IV. Experimental Results and Discussion

The existing design and proposed design are simulated using ISE simulator and verified for its functionality and then synthesized using the Cadence GENUS EDA tool for the technology node of 65nm using TSMC 65nm library.

The synthesis result shows the Power, Area and time delay of each design. By analyzing the results, it is observed that the design proposed has lesser power and area compared to existing design but with some reasonable increase in the delay.

The simulation and synthesis is done for both Dadda Multiplier with proposed full adder and ALU, the results of the experiment are as follows

1. 16-bit Dadda Multiplier with conventional full adder

Simulation results for verifying the functionality and synthesize results of existing 16-bit Dadda Multiplier with Conventional full adder and CLA is shows in fig 6.

16-bit Dadda multiplier is synthesized using TSMC 65nm library using Cadence GENUS EDA tool and synthesis results show the power and area of 223799.941nW and 2442.24um^2 respectively, with the timing delay of 3776 ps.

2. 16-bit Dadda Multiplier with optimized full adder

Simulation results for verifying the functionality and synthesize results of existing 16-bit Dadda Multiplier with Conventional full adder and CLA is shows in fig 7.

16-bit Dadda multiplier is synthesized using TSMC 65nm library using Cadence GENUS EDA tool and synthesis results show the power and area of 189500.943nW and 2359.44um^2 respectively, with the timing delay of 4163 ps.

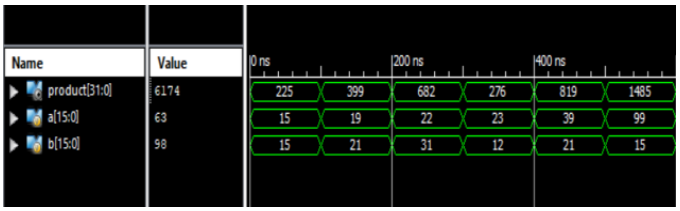

Fig.6 Simulation results of 16-bit Dadda multiplier with conventional full adder

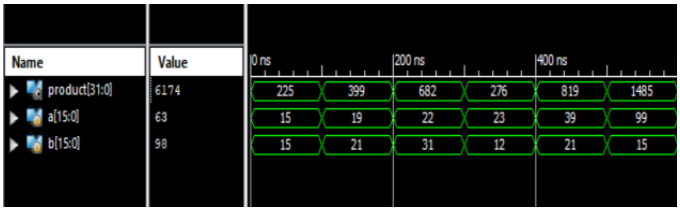

Fig.7 Simulation results of 16-bit Dadda Multiplier with optimized full adder

3. ALU with Dadda Multiplier having conventional full adder

Simulation results for verifying the functionality and synthesize results of ALU with Dadda Multiplier with Conventional full adder and CLA is shows in fig.8

16-bit ALU design with existing dadda Multiplier and carry look-ahead adder is synthesized using TSMC 65nm library using Cadence GENUS EDA tool and synthesis results show the power and area of 333536.34nW and 4429.08um^2 respectively, with the timing delay of 17301 ps.

4. ALU with Dadda Multiplier with optimized full adder and carry look ahead adder

Simulation results for verifying the functionality and synthesize results of ALU with Dadda Multiplier with Conventional full adder and CLA is shows in fig.9.

16-bit ALU design with existing dadda Multiplier and carry look-ahead adder is synthesized using TSMC 65nm library using Cadence GENUS EDA tool and synthesis results show the power and area of 264634.66nW and 4349.16 um^2 respectively, with the timing delay of 17301 ps.

The Table III represents the synthesized results showing the power, area and timing report of the 16-bit Dadda multiplier with conventional full adder and proposed full adder. 16-bit ALU results implemented with Dadda multiplier with conventional full adder and proposed full adder.

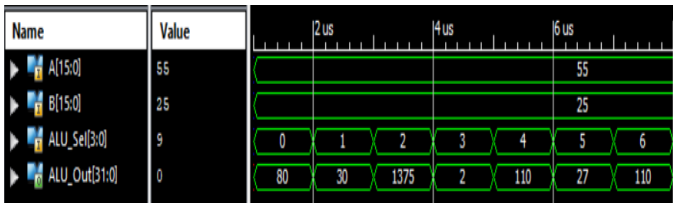

Fig.8 Simulation results of Arithmetic and Logical Unit

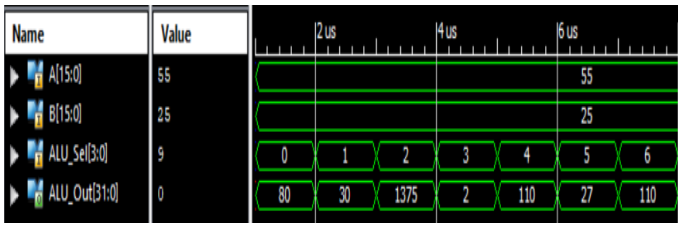

Fig.9 Simulation results of Arithmetic and Logical Unit with proposed dada multiplier

Table III Synthesized results the design models

Design	Power (nW)	Area (um^2)	Timing (ps)
Dadda Multiplier [8]	679000	1954	6.278
Dadda multiplier with Conventional full adder	223799.941	2442.24	3776
Dadda multiplier with Proposed full adder	189500.937	2395.44	4163
ALU with existing Dadda multiplier	333536.341	4429.08	17301
ALU with modified Dadda multiplier having proposed full adder	264634.657	4349.19	17301

V. CONCLUSION

In order to reduce the total power dissipation in VLSI design, numerous approaches has been introduced each one has its own level of power reduction. Here, the technology mapping techniques are applied for designing new low power data-path architecture which intern reduces the overall power dissipation.

In the proposed research the new architecture for the full adder design is implemented with technology node of 65nm. This full adder is implemented to 16-bit dadda multiplier and the synthesis results are obtained. From the above synthesis results, the 16-bit Dadda Multiplier and ALU with proposed full adder has minimum power and area efficient. Here it is observed that the proposed design has lesser power and area than the Dadda Multiplier and ALU having conventional full adder respectively. The conventional full adder has the total power dissipation of 223.8µW and area of 2442.24um^2. The proposed design has the power and area of 189.5µW and 2395.44um^2 respectively. From the result it is clear that the proposed multiplier design improved power dissipation and area of 15.32 and 1.92% respectively, when it is compared the dada multiplier having conventional full adder. Similarly, when proposed multiplier having low power adder architecture is implemented to 18bit Arithmetic and Logical Unit the power and area improvement are given by 20.34% and 1.80%, respectively. Since VLSI always has the trade-off between Area Power and Timing and for ALU 20.34% and 1.80% power and area lesser than the existing ALU respectively. The concept of optimized full adder is implemented in multiplier, ALU and it is also extended to other VLSI chip designs.

REFERENCES

[1] Muhammad Hussnain Riaz, "Low power 4×4 bit multiplier design using dadda algorithm and optimized full adder", 15th international Bhurban conference, 2018.

[2] Ashish KumarYadav, "Low power high speed 1-bit full adder circuit design at 45nm cmos technology", Proceeding International conference on Recent Innovations is Signal Processing and Embedded Systems, ISBN 978-1-5090-4760-4/17/©2017 IEEE) ,2017

[3] Zain Shabbir, Anas Razzaq Ghumman, Shabbir Majeed Chaudhry, "A reduced-sp-d3lsum adder-based high frequency 4 × 4 bit multiplier using dadda algorithm", Springer Science and Business Media New York 2015.

[4] R.Abhilash, Sanjay Dubey,Chinnaaiah.M.C "ASIC design of low power vlsi architecture for different multiplier algorithms using compressors", International Conference on Industrial and information Systems, ICIIS, 2016.

[5] B. Ramkumar, V. Sreedeep and Harish M Kittur, "A design technique for faster dadda multiplier" Member, IEEE,

[6] Mr. M. Merlin Moses, "Design of high speed and low power dadda multiplier using different compressors", Asian Journal of Applied Science and Technology (AJAST) (Open Access Quarterly International Journal) Volume 2, Issue 2, Pages 419-424, April-June 2018.

[7] Assem Hussein, "A 16-bit high-speed low-power hybrid adder", IEEE,2016.

[8] S. Ravi, Govind Shaji Nair, "Low power and efficient dadda multiplier". Research Journal of Applied Sciences, Engineering and Technology 9(1): 53-57, 2015.

[9] S.Srikanth, "Low power array multiplier using modified full adder", 2nd IEEE International Conference on Engineering and Technology (ICETECH), 17th and 18th March 2016, Coimbatore, TN, India.

[10] K. Anirudh Kumar Maurya, "Design and implementation of 32-bit adders using various full adders", International Conference on Innovations in Power and Advanced Computing Technologies [i-PACT2017] .

[11] Himani Upadhyay and shubhajit Roy Chowdhury, "Design of high speed and low power 5:3 compressor architecture using novel two transistor xor gates", CVEST,2014.

[12] Akhil M.S and Muthukumaran. N, "Design of optimizing adders for low power digital signal processing", International Journal of Engineering Research and Applications, Vol. 5, pp. 59-65, March 2014.

[13] Yingtao Jiang, Abdulkarim Al-Sheraidah, Yuke Wang, Edwin Sha, and Jin-Gyun Chung, "A novel multiplexer-based low-power full adder", IEEE Transactions on Circuits and Systems-II: Express BRIEFS, vol. 51, no. 7, July 2004.

[14] I. S. Abu-Khater, A. Bellaouar, M. I. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers," IEEE Journal Solid State Circuits, vol. 31, no. 10, pp. 1535–1546, Oct. 1996.

[15] Maroju SaiKumar and Dr. P. Samundiswary, "Design and performance analysis of various adders using verilog" in International Journal of Computer Science and Mobile Computing, pp.128-138,Vol.2, Issue 9, September 2013.

[16] Yu Zhou and Hui Guo, "Application specific low power alu design", IEEE/IFIP International Conference on Embedded and Ubiquitous Computing,2008.

[17] Prakhi Agrawal, "A high speed binary floating point multiplier using dadda algorithm", IJRCCE, May 2015.

[18] Vijayalakshmi Bandi, "Performance analysis for vedic mutliplier using modified full adder", International Conference on Innovations in Power and Advanced Computing Technologies [i-PACT2017].

[19] R. Rajeswari, "Design and analysis of various standard multipliers using low power very large scale integration (VLSI)", International Journal of MC Square Scientific Research Vol.4, No.1 Nov 2012.

[20] R.Prathiba, "Design of high performance and low power multiplier using modified booth encoder", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) – 2016.