

# Design and Implementation of Compressor based 32-bit Multipliers for MAC Architecture

M. Abheesh Kumar, A. Sudhakar, J. Venkata Suman

**Abstract:** Arithmetic operations play a major role in digital circuit design like adders, multipliers etc. Multiplication is an important fundamental arithmetic operation in high performance systems such as microprocessor and digital signal processors circuits. Implementation of multipliers using compressor circuit over conventional adders will reduce the number of levels of addition, which will in turn reduces the latency of the multiplier. Multiplier module is most likely the essential part of MAC (Multiplier-Accumulator) unit design. Compressor based multipliers in MAC architecture design results high performance. FPGA and ASIC implementations of 4:2 compressor based 32-bit Wallace and Dadda multipliers can be done by using Xilinx Vivado and Cadence CMOS technology tools. These results are compared with other multiplier designs with respect to area, latency and power dissipation.

**Keywords:** Compressors, Verilog HDL, Cadence CMOS 90nm technology, Xilinx Vivado, MAC.

## I. INTRODUCTION

Now a day's, the electronic technological developments of the present world are changing rapidly in construction of small size, less power and lower delay devices. The processing speed of the device is improved due to limiting the power dissipation [3]. A survey had been carried out with different multiplier algorithms for MAC architecture design [7-8]. Usage of the traditional shift and add algorithm in implementation of multiplier design is not suitable in delay point of view [9]. To overcome the drawbacks in traditional shift and add algorithm some parallel multiplier structure is incorporated to perform multiplication. Tree based multipliers like Binary tree and Wallace tree, array based multipliers such as Braun, Booth and Baugh Woolley are some of the parallel multipliers [2][9]. It is shown that the parallel multipliers are faster than the traditional multipliers, Wallace and Dadda multipliers come under parallel multipliers [3]. In general, for  $n \times n$  bit multiplication process an array of  $n$ -AND gates required to generate partial products in partial production generation stage, an array of  $n^2$  adders (adder may be full adder or half adder) to sum the  $n^2$  partial product terms for partial product accumulation stage and addition stage. Wallace tree multipliers comes under parallel multiplier because the partial product reduction done in parallel way. A Wallace tree multiplier consists of several stages of partial product reduction depending on multiplier size. In each stage, the accumulation of partial products process evaluated by the different adder architectures like full adder, half adder etc [2]. In the preliminary stages of the

Wallace tree formation, the partial products do not depend upon inputs acquired from the AND array but not on any other values like carry. From the advance higher stages, the final value includes the carry out value generated in preceding stage. This process is repetitive for the remaining stages of multiplier structure [11-12]. Dot structures of Wallace and Dadda multipliers are shown in Figure 1 and Figure 2 respectively.



Figure 1: Dot structure for Wallace tree multiplier

In Dadda multipliers, column wise addition is performed. Formation of partial products stage, partial products obtained with the help of an array of AND gates by performing AND operation between multiplier and multiplicand. After that, partial products columns are arranged as inverted delta structure form by shifting upwards. After formation of an inverted delta structure of partial products, reductions of partial products are done by using suitable combinational adder architecture [7]. In this paper a compressed based multiplication is used in Wallace tree and Dadda architectures. So that speed is increased considerably.

Revised Manuscript Received on July 06, 2019.

M. Abheesh Kumar, PG scholar, GMR Institute of Technology, India.

A. Sudhakar, Associate Professor, Department of ECE, GMRIT, India.

J. Venkata Suman, Assistant Professor, Department of ECE, GMRIT, Rajam, India.

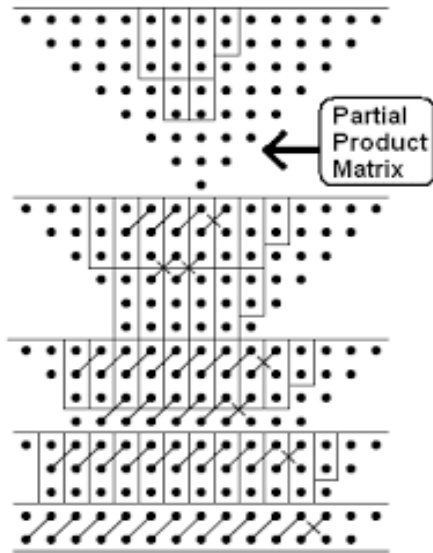


Figure 2: Dot structure for Dadda multiplier

II. METHODOLOGY

In Boolean algebra, half adder and full adder circuits are used in multipliers to perform partial product summation. But in multipliers, huge numbers of adders are required to perform summation and each adder carry is propagated to the next adder as an input. Hence the delay between these adders shows an abrupt result in critical path. To optimize latency, power, area and critical path, optimized designs introduced in place of conventional adders. Multiplexer based adder circuits [5], parallel prefix adders [6][13] and compressors are better replacement for the conventional adder circuits to acquire better results [1][4].

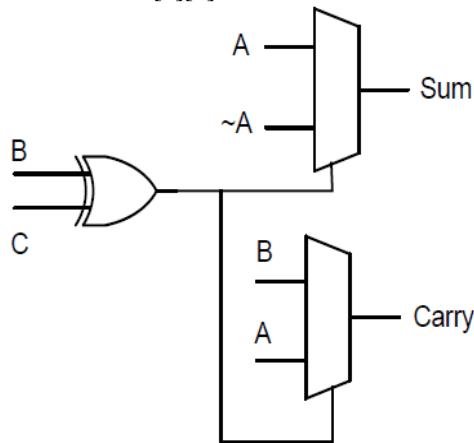


Figure 3: Multiplexer based full adder design

Output generated by the XOR gate is taken as input to the selection line for the multiplexer modules. Adder outputs (Sum and Carry) are produced depending on the input given to the selection line and multiplexer inputs. These adder modules are used as sub modules in our required design. Kogge-Stone adder concept of basic parallel prefix adder was first time implemented and designed by M. Kogge and S. Stone [6][13]. Three phases are there in KSA addition, they are preprocessing stage, carry generation stage and post processing stage. Preprocessing stage of the KSA, propagated signal and generated signal are calculated as like as carry look ahead adder. Carry generation stage is a most major block in KSA design. It consists of two components such as black cell and gray cell. Gray cell is used to generate generated signal

alone and this signal used in the calculation of the sum in subsequent stage. Black cell produces the generated signal and propagated signal, required to the calculation of the subsequent phase. Post processing is the final stage of the adder; final sum evaluation is the final outcome of the post processing stage [14]. The 2-bit KSA architecture is shown in Figure 4. In this paper, we extended this logic to 16-bit KSA and further.

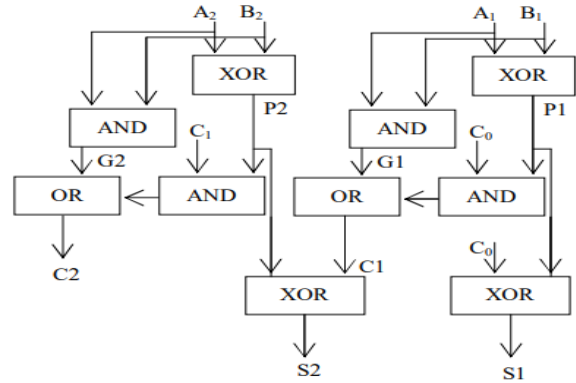


Figure 4: 2-bit KSA Architecture

A compressor circuit in VLSI compresses the number of outputs for a given input. Here the main inputs are bits to be added and outputs are sum along with carry.

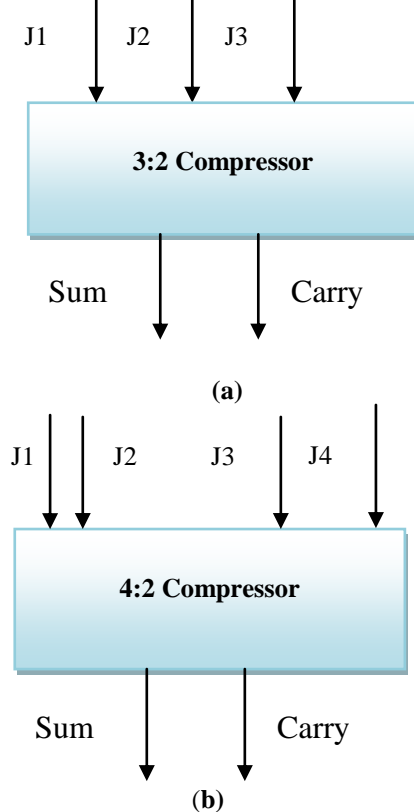


Figure 5: Compressor architectures a) 3:2 b) 4:2

In 3:2 compressors inputs J1, J2 and J3 are given and generated outputs are sum and carry. A 4-2 compressor inputs be J1, J2, J3, J4 and Carry\_in the produced outputs be Sum, Carry along with Carry\_out. 3:2 and 4:2 compressors are shown in Figure 5. Internal architecture of 4:2 compressors with XOR gates and 2:1 multiplexers (MUX) modules is shown in Figure 6. The multiplexer based 4:2

compressors is advantageous to the full adder based 4:2 compressors, we used multiplexer based 4:2 compressors in our multipliers design. It includes a critical path delay of three XOR gates.

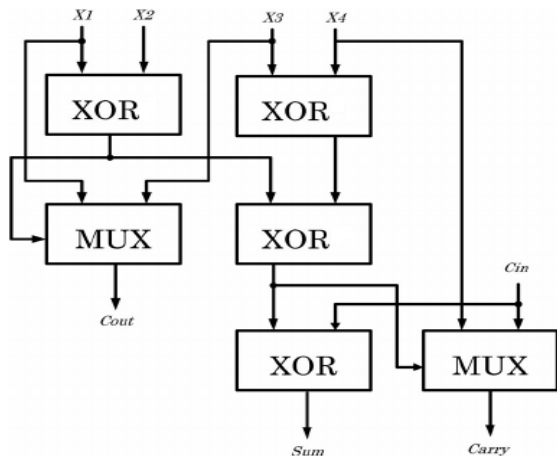


Figure 6: 4:2 compressor design using XOR-MUX

Wallace tree multiplier design with 4:2 compressors and KSA architecture is shown in Figure 7.

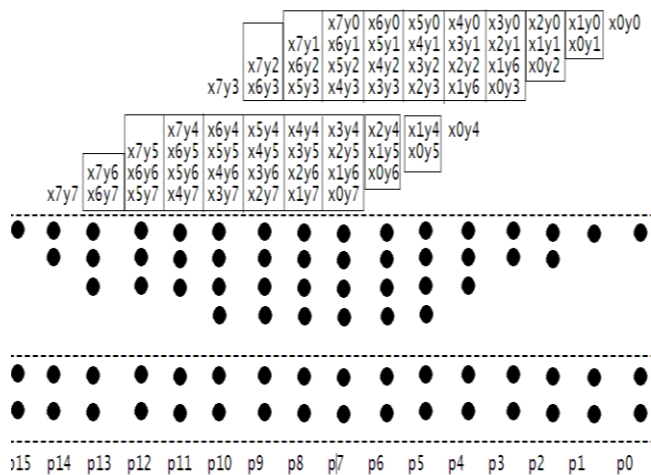


Figure 7: Wallace multiplier using 4:2 compressors

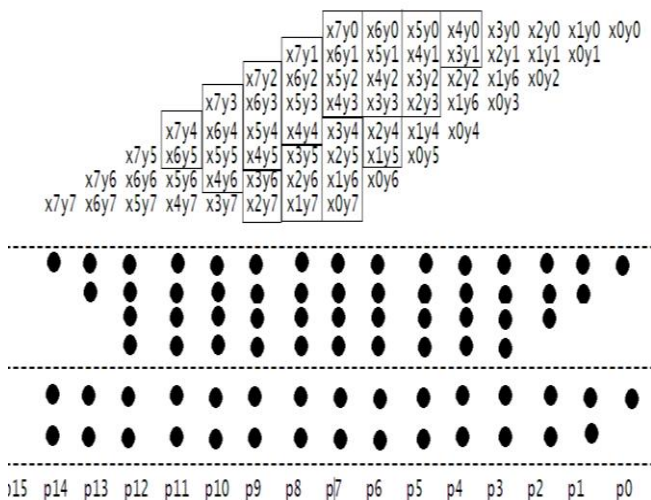


Figure 8: Dadda multiplier using 4:2 compressors

Dadda tree multiplier design with 4:2 compressors and KSA architecture is shown in Figure 8. Further the range of

multiplier is increases from 8\*8 to 16\*16 multiplier, which gives a resultant of 32-bit multiplier as shown in Figure 9.

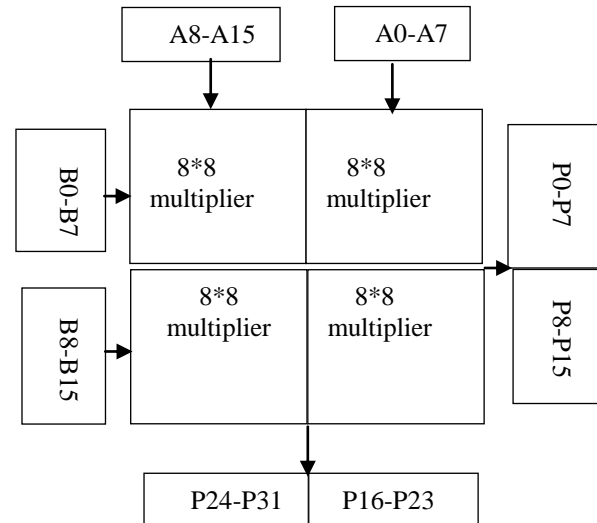
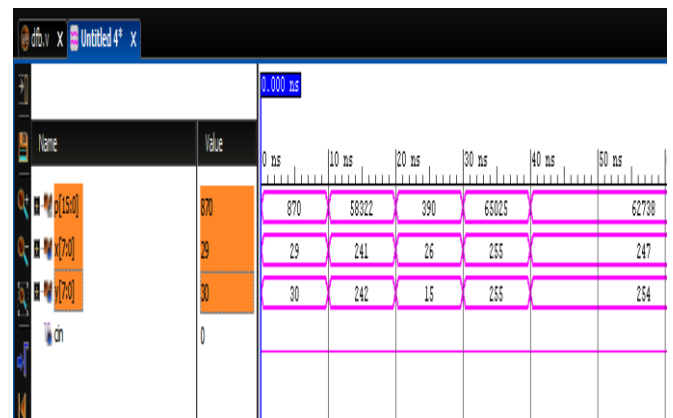


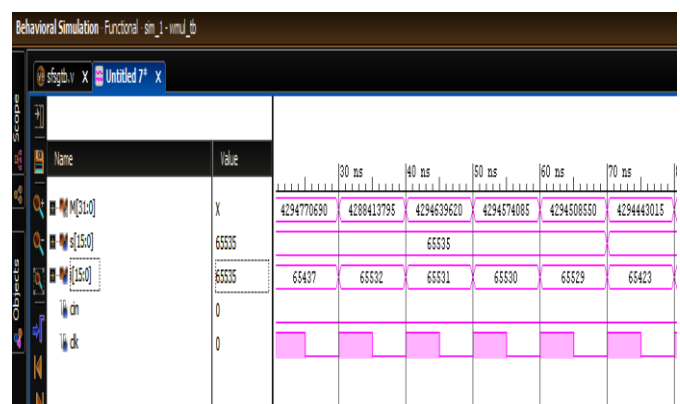
Figure 9: Design of 16\*16 multiplier from 8\*8 multiplier

### III. RESULTS AND DISCUSSION

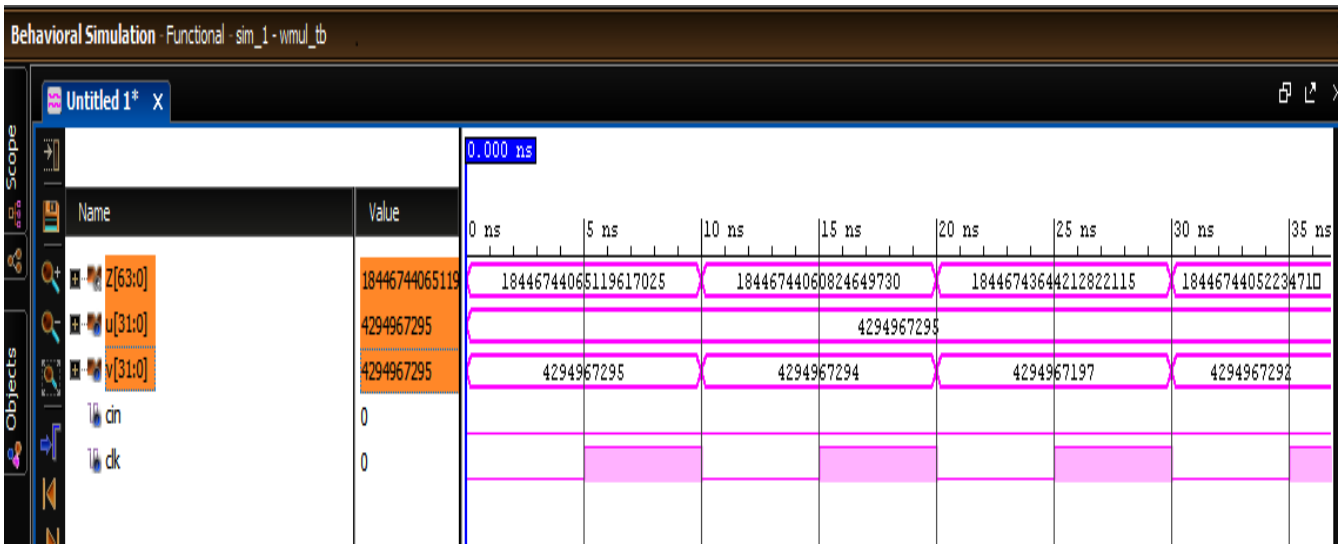
Multiplier design is written in Verilog HDL and simulated using Xilinx Vivado 16.2 simulation tool. The simulation results for 8-bit, 16-bit and 32-bit multipliers are shown in Figure10.



a) 8-bit simulation results



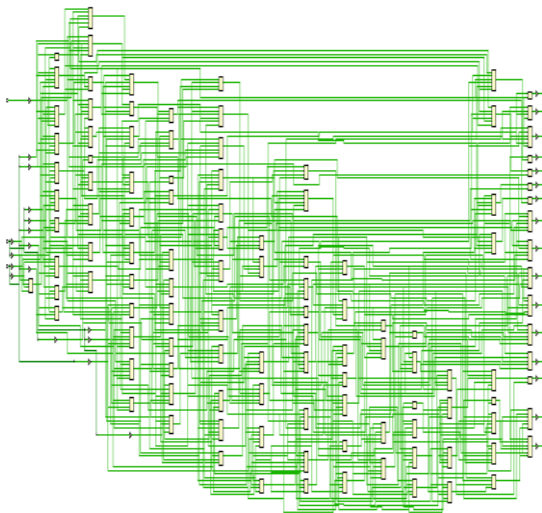
b) 16-bit simulation results



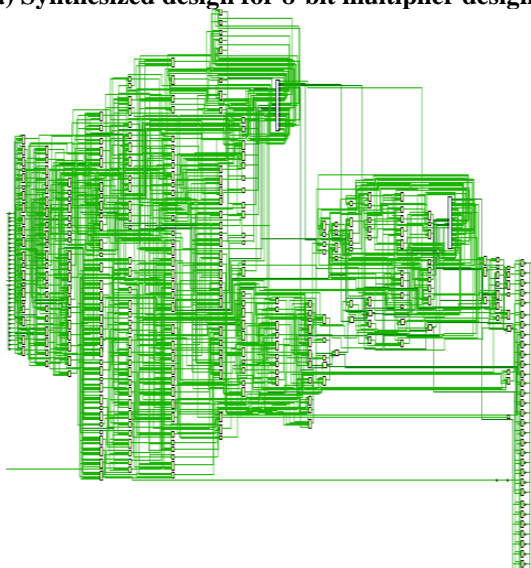
c) 32-bit simulation results

Figure 10: Simulation results a) 8-bit, b) 16-bit and c) 32-bit

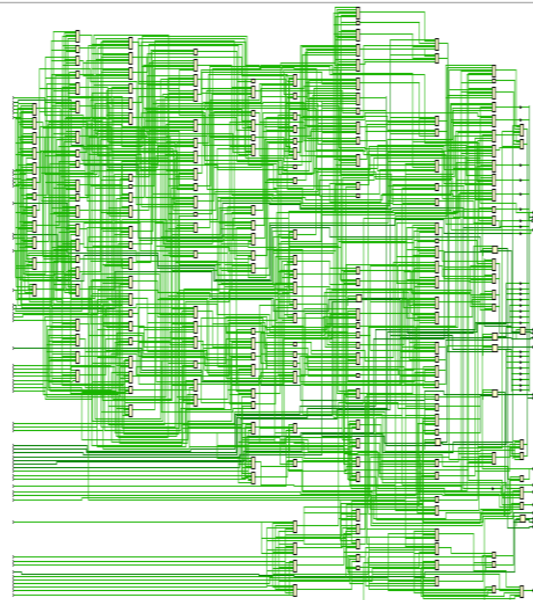
The net list design or synthesized design of multipliers using Xilinx is shown in Figure 11.



a) Synthesized design for 8-bit multiplier design



b) Synthesized design for 16-bit multiplier design



c) Synthesized design for 32-bit multiplier design

Figure 11: Synthesized designs for multiplier a) 8-bit b) 16-bit c) 32-bit

All the multiplier designs are implemented using Zynq evaluation development kit (ZED board) with the targeted device xc7z020dg484-1 FPGA kit. The Table 1 gives the information of 8\*8 bit, 16\*16 bit, 32\*32 bit multipliers utilization reports. A cell utilization report for the design consists of LUTs count, IO along with flip flops.

# Design and Implementation of Compressor based 32-bit Multipliers for MAC Architecture

**Table 1: FPGA results of proposed multiplier designs**

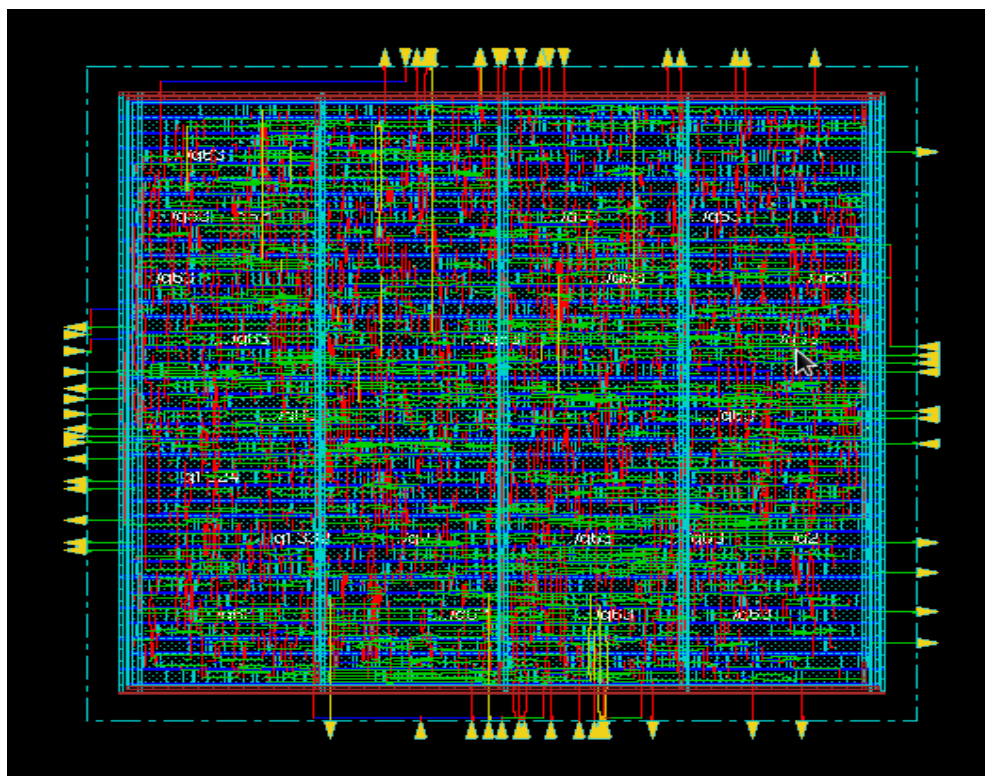
Multiplier type	Used LUT's count out of available 53200		
	8*8	16*16	32*32
Wallace multiplier using Full adder (Normal)	101	381	1656
Wallace multiplier using 3:2 compressor	93	345	1634
Wallace multiplier using 4:2 compressor (FA basic)	87	393	1608
Wallace multiplier using 4:2 compressor (FA designed using MUX)	94	420	1767
Wallace multiplier using 4:2 compressor using mux	85	429	1743
Dadda multiplier using 4:2 compressor (FA basic)	88	360	1471
Dadda multiplier using 4:2 compressor (FA designed using MUX)	96	381	1587
Dadda multiplier using 4:2 compressor using mux	103	395	1649

Table 2 provides information about the comparison of LUT count of proposed design with other designs using Xilinx Spartan-3E FPGA kit with device 3S100EVQ100-5.

**Table 3: Performance comparison of multipliers**

Types of 16-bit Multiplier	Used LUT's count out of available 1920
Dadda [3]	889
Wallace [3]	1000
Proposed Wallace	619
Proposed Dadda	576

ASIC implementations of all the multiplier designs are done by Cadence Semi-custom standard cell design using TSMC 90nm libraries. NCSIM, RC compiler and Encounter tools for simulation, synthesis and implementation phase of design respectively. Figure 12 includes the implementation of the multiplier designs with different adder (sub module) designs. After the physical design step, we get GDSII file for further hardware implementation or fabrication of the design.



**Figure 12: ASIC implementation of proposed multiplier**

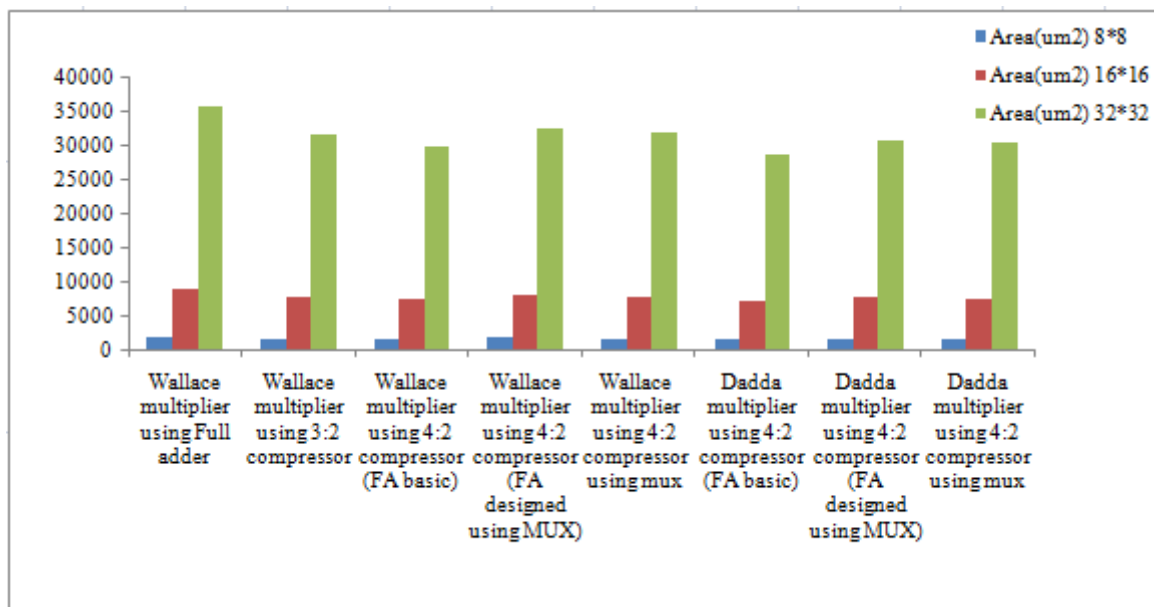
The Table 3 gives the information of 8\*8 bit, 16\*16 bit, 32\*32-bit multipliers synthesis reports for area, power and delay.

Table 3: ASIC Synthesis results of proposed multiplier designs

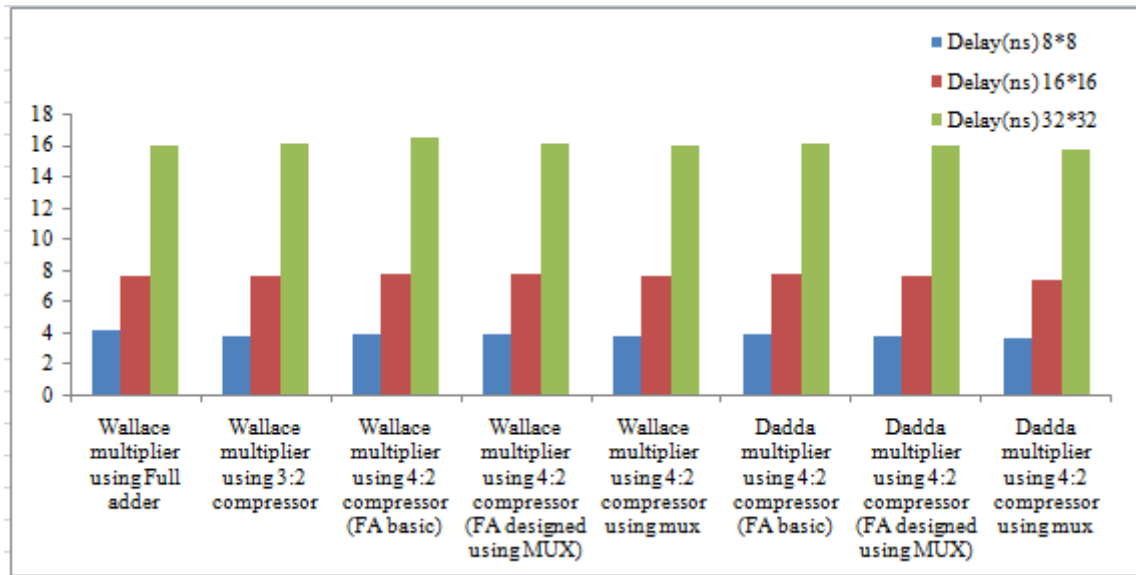
Multiplier type	Area ( $\mu\text{m}^2$ )			Delay (ns)			Power ( $\mu\text{W}$ )		
	8*8	16*16	32*32	8*8	16*16	32*32	8*8	16*16	32*32
Wallace multiplier using Full adder	1845	8729	35589	4.087	7.550	15.914	87.794	521.35	2298.36
Wallace multiplier using 3:2 compressor	1596	7733	31604	3.775	7.646	16.009	97.721	567.68	2476.59
Wallace multiplier using 4:2 compressor (FA basic)	1483	7282	29800	3.840	7.688	16.502	78.759	514.34	2289.14
Wallace multiplier using 4:2 compressor (FA designed using MUX)	1642	7918	32416	3.895	7.714	16.078	99.929	604.66	2662.23
Wallace multiplier using 4:2 compressor using mux	1605	7773	31762	3.767	7.604	15.968	98.273	599.26	2611.62
Dadda multiplier using 4:2 compressor (FA basic)	1398	6943	28444	3.895	7.659	16.023	72.361	467.71	2107.46
Dadda multiplier using 4:2 compressor (FA designed using MUX)	1540	7512	30720	3.786	7.560	15.924	90.156	541.46	2398.82
Dadda multiplier using 4:2 compressor using mux	1514	7406	30296	3.622	7.351	15.715	92.872	550.30	2436.98

From the above synthesis results of different multiplier architectures, we can conclude that the multiplier design using basic compressor circuits designed using full adders requires less area and power in both Dadda and Wallace tree multipliers. At the same time the designs with multiplexer

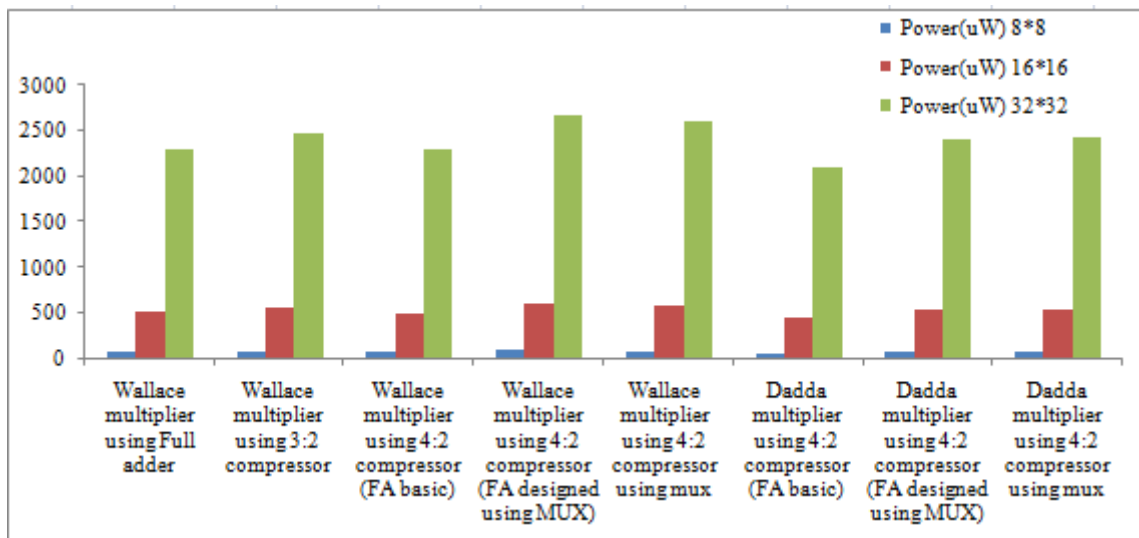
based compressor circuits providing less delay in both Dadda and Wallace tree based multiplier designs. Proposed multipliers ASIC synthesis design reports for area, power and delay are shown as bar graph in Figure 13.



a) Area reports



b) Delay report



c) Power report

Figure 13: ASIC synthesis reports in bar graph a) Area report, b) Delay report and c) Delay report

Table 3 provides information about the comparison of various factors of proposed design with other designs. From this comparison table the proposed multiplier designs posse’s area of nearly 50% lesser, 40% lesser power improvements.

Table 3: Comparison of Proposed Multiplier with Other Multipliers in the Literature

Reference	Technology (nm)	Size	Area (um <sup>2</sup> )	Power (mW)	Delay (ns)
[5]	SAED 90	8-bit	2701	1.106	4.44
[11]	Synopsys 90	8-bit	3262	1.94	2.64
[11]	Synopsys 90	8-bit	3148	1.96	2.66
Proposed Wallace (basic compressor)	TSMC 90	8-bit	1483	0.078	3.84

Proposed Wallace (multiplexer based compressor)	TSMC 90	8-bit	1605	0.098	3.7
Proposed Dadda (basic compressor)	TSMC 90	8-bit	1398	0.72	3.89
Proposed Dadda (multiplexer based compressor)	TSMC 90	8-bit	1514	0.092	3.62

#### IV. CONCLUSION

Wallace multiplier and dada multiplier design by using compressor circuits, we observed that the reduction in critical path delay, area and power. This results higher circuit speed at low power and area. At the same instant, the usage of adders in the design are limited so that reduction in the complexity of the circuitry. In some cases, a nominal increase in the area but for the area delay product for that design becomes lower. Kogge-Stone adder usage in the final stage addition provides low latency in the design. From this analysis, 4:2 compressors combination with the parallel prefix adders surely provide faster operation in multiplier-based designs at lower area and power. These multipliers can be readily replacement for situations where speed critical applications in VLSI. In future work, further improvements of the multipliers can be done by implementing new sub modules designs in adders and compressors.

#### REFERENCES

1. Abdoreza Pishvaie, Ghassem Jaberipur and Ali Jahanian, "High performance CMOS (4:2) compressors", *International Journal of Electronics*, vol. 101, Issue 11, pp. 1511-1525, 2014.
2. C. S. Wallace, "A suggestion for a fast multiplier", *IEEE Transactions on Electronic Computers*, vol. EC-13, Issue 1, pp. 14-17, 1965.
3. G. Challa Ram, D. Sudha Rani, R. Balasaikesava and K. Bala Sindhuri, "Design of delay efficient modified 16 bit Wallace multiplier", *IEEE International Conference On Recent Trends In Electronics Information Communication Technology*, 2016.
4. J. Tonfat and R. Reis, "Low power 3-2 and 4-2 adder compressors implemented using ASTRAN", *IEEE 3rd Latin American Symposium on Circuits and Systems*, pp. 1-4, 2013.
5. K. B. Jaiswal, V. Nithish Kumar, P. Seshadri and G. Lakshminarayanan, "Low power Wallace tree multiplier using modified full adder", *3rd International Conference on Signal Processing, Communication and Networking*, Chennai, pp. 1-4, 2015.
6. Lee Mei Xiang, Muhammad Mun'im Ahmad Zabidi, Ainy Haziyah Awab and Ab Al-Hadi Ab Rahman, "VLSI implementation of a fast Kogge-Stone parallel-prefix adder", *International Post Graduate Conference on Applied Science & Physics*, 2017.
7. P. A. Irfan Khan and Ravi Shankar Mishra, "Comparative Analysis of different Algorithm for Design of High-Speed Multiplier Accumulator Unit (MAC)", *Indian Journal of Science and Technology*, vol. 9, Issue 8, 2016.
8. P. Jebashini, R. Uma, P. Dhavachelvan and Hon Kah Wye, "A survey and comparative analysis of Multiply-Accumulate (MAC) block for digital signal processing application on ASIC and FPGA", *Journal of Applied Sciences*, vol. 15, Issue 7, pp. 934-946, 2016.
9. R. Abhilash, S. Dubey and M. C. Chinnaiah, "Asic design of low power VLSI architecture for different multiplier algorithms using compressors", *11th International Conference on Industrial and Information Systems*, pp. 387-392, 2016.
10. R. Abhilash, S. Dubey and M. C. Chinnaiah, "ASIC design of signed and unsigned multipliers using compressors", *International Conference on Microelectronics, Computing and Communications*, pp. 1-6, 2016.
11. Shahzad Asif and Yinan Kong, "Low-area Wallace multiplier", Department of Engineering, Macquarie University, Sydney, NSW 2109, 2014.
12. S. N. Kumar et al, "Design of an area-efficient multiplier", *International Conference on Recent Advances in Electronics and Communication Technology*, pp. 329-332, 2017.
13. P. Ramanathan, P. Kowsalya and P. Anitha, "Modified low power Wallace tree multiplier using higher order compressors", *International Journal of Electronics Letters*, vol. 5, Issue 2, pp. 177-188, 2017.
14. Vivek Gupta and Vaibhav Jindal, "VLSI architecture for complex vedic multiplier using hybrid square Kogge-Stone adder technique", *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 7, Issue 6, pp. 7513-7520, 2018.

#### AUTHORS PROFILE



**M. Abheesh Kumar** received his Bachelor of Technology in Electronics and Communication from Sasi Institute of Technology and Engineering in the year 2014. His areas of interest include Low power digital VLSI design.



Communication.

**Dr. A. Sudhakar** received his Ph. D. from Jaipur National University, Jaipur in the year 2016. Presently he is working as an Associate Professor in ECE department at GMR Institute of Technology, Rajam. He has published/presented 15 papers in National and International Journals/ Conferences. His main research interest includes antennas and Wireless



Technology, Rajam. His areas of interest include Low power VLSI design and Signal Processing.

**Jami Venkata Suman** received his Bachelor of Engineering in Electronics and Communication from Visvesvaraya Technological University, Karnataka state in the year 2004 and Master of Technology in VLSI System Design from JNTUH, Hyderabad in the year 2008. He is currently working as an Assistant Professor in the Department of Electronics and Communication Engineering at GMR Institute of