Design and Development of Concurrent Vedic Architecture using ASIC Design Flow

Dinubhau B. Alaspure[1], Dr. Swati Rajesh Dixit[2], Dr. Jitendra S. Edle[3]

[1]Research Scholar,
Department of Electronics and Telecommunication Engineering
G. H. Raisoni University, Amravati
Amravati, India

[2]Department of Electronics and Telecommuinication Engineering
G. H. Raisoni Institute of Engineering and Technology, Nagpur
Nagpur, India
Assistant Professor

[3]Department of Electronics and Telecommuinication Engineering
Sipna College of Engineering and Technology, Amravati
Amravati, India

Abstract— The proposed research paper discloses the design and development of the Concurrent Vedic Multiplier Architecture using ASIC design flow. In this development process, the traditional Vedic principles are referred for the design of the multiplier architecture. The principles are converted into the Boolean statements and described using HDL language. The development process is further carried out through simulation, synthesis, RTL extraction, tight optimization and analysis. Different variants of the Xilinx FPGAs are preferred for concurrent implementation and ASIC design flow is then deployed for tight optimization using Cadence tool.

Keywords— Vedic Mathematics, Virtex, Kintex, Artix, VHDL.

Introduction

Multiplication is the basic mathematical computation which we perform in daily life for simple applications like calculating the exchange rate to the spending money in holidays and it is also involved in the high-tech applications like scientific computing to the pattern recognition, face recognition to the image processing and signal processing and tones of other applications. Out of all other tiny operations multiplication is the most hardware intensive operation when it is implemented on the hardware platform.

In this research paper, an attempt is made to check the impact on the hardware intensity of the multiplier architecture, if Vedic fundamentals are deployed instead of the traditional other approaches. To have the sophisticated performance of the development, directly, the architecture is described using the hardware description language. The architecture is time-simulated, and the performance is assessed for logical correctness of the design. In the next stage of the development, the architecture is synthesized by targeting the architecture to the different variants of the Field Programmable Gate Array (FPGA). The FPGA are most suitable for development of the concurrent architecture. The through synthesis the RTL verification is carried out through RTL synthesis tool and the statistical analysis is carried with respect to the different parameters like time, frequency, power, and resource utilization. In the last step of the development, implementation process is carried out. Implementation is comprising of different tiny steps like place, map, and route through which completes the FPGA design flow. For functional verification and synthesis process, the Xilinx Vivado tool is preferred.

While analysing the outcomes in terms of the time, power, resource utilization and frequency it is observed that further, next level of optimization is possible if the applications specific architecture of the proposed architecture is realized. Accordingly, the same HDL description is executed through the ASIC design flow using Cadence Tool. Highly improved outcomes are observed after execution through the ASIC design flow. In the subsequent section, the detailed literature review, statistical outcomes through FPGA design flow and outcomes after ASIC design flow is discussed in depth.

Literature Review

In this paper, the authors [1] have proposed and implemented an 8-bit Vedic multiplier and contrast its performance with several traditional Vedic multipliers such as Array, Booth and Wallace tree multiplier. Vedic computations are the process of mathematical calculations to execute the multiplication of two binary numbers. The proposed work used vertical and crosswise Vedic sutra for the implementation of multiplier which gives improved performance and utilizes a lesser amount of time for calculation. To execute the addition of partly produced multiplications, a Modified carry save adder is utilized. It decreases the execution delay towards the summation of uncompleted multiplications. Verilog hardware description language is used to develop the proposed algorithm design. The proposed algorithm is also authenticated on a field programmable gate array platform. The output shows that the proposed work is effective and utilizes 13.245 nanoseconds for the multiplication operation. The systematic block arrangements referred through the above cited paper is disclosed through the subsequent figure.

The structure of the multiplier and accumulate unit can be realized by employing the Vedic multiplier with the reversible gates. Out of available 16 Vedic sutras, the Vedic multiplier is widely realized using the Urdhva Tiryagbhyam sutra. To attain the high speed of operation, the multiplier and adders can be realized by using reversible logic gates. A Vedic multiplier gives higher performance, lessens region and decreases the partial multiplications. This paper proposed four Vedic multipliers such as 8-bit, 16-bit, 32-bit and 64-bit designed utilizing DKG adder, kogge stone adder and carry save adder. Out of the proposed three adder scheme it is found that the DKG gate with Vedic multiplier adder is having a high speed of operation. [2]

As the appearance of handheld, portable electronic gadgets which runs on batteries increases day by day, the necessity of designing low power and high-speed circuit has become a challenging task. Very large scale integrated are focused on various technologies to decrease the power utilization and how to improve system performance. In all execution processes multiplication is an essential operation. Realization of an effective multiplier with low power consumption and improved performance leads to devices with lessen power utilization and high speed. Vedic multiplier when employed with reversible logic gates there is no power dissipation in the circuit. The authors proposed a system with Vedic multiplier and reversible logic gates assemble to execute complex multiplication with enhanced performance and utilizing very little power. Vedic multiplier based on Urdhva Tiryagbhyam sutra is used to perform high speed multiplication computation. [3]

Multiplication is a fundamental arithmetic process used in several very large-scale integrated structures such as high-speed circuits, digital signal processors, multipliers and accumulates unit architectures and microprocessor structures. Recurrent utilization of logical gates and arithmetic operations in the application of information processing, information transmission, accuracy plays a vital task. In this paper, the authors proposed an error resistant Vedic multiplier that is built using error resistant logic gates. As the proposed technique neglects the soft errors, the results generated by this circuit are fully trustworthy. In the existence of error, a repair signal is initiated to present accurate results. This is attained at a cost of hardware duplication however the crucial pathway remains identical to that of the conservative system with no error repair circuitry. [4]

In today's era, digital applications like digital signal processing, multimedia, 3-dimensional graphic designs and diversity of scientific computations necessitate critical multiplications computations, which are implemented by the advanced microprocessor executing element. To deal with such critical computations, a fixed-point number is not sufficient as it has a defined number of bits for the integer part and defined bits for the decimal part. In such conditions, a floating-point number system is employed to deal with complex computations. This paper proposed the Vedic mathematics based low power and region proficient four terms fused dot multiplication unit. To improve the performance of the system, the proposed floating point multiplication unit utilized the Vedic multiplier, leading null expectancy, normalization, rounding off and compound addition. To assess the performance of the proposed design, the region and power are computed. Also, the proposed design is implemented for a specific accuracy floating point multiplier. [5]

Vedic mathematics is a branch of Sthapatya Veda which is an important supplement of Atharva Veda. Bharti Krishna Teerthaji known as the father of Vedic mathematics studied for several years and assembled all his research collectively. He developed Sutras that consists of 16 formulas and Upasturas that again consists of 16 sub formulas. The authors [6] proposed a novel squaring technique approach to made complex computations simpler and the circuits more effective. The proposed Vedic mathematics technique is employed to perform the squaring of two binary numbers. The result shows that time required for squaring calculation was decreased by utilizing Vedic mathematics. The proposed Vedic structure demonstrated of lesser number of transistors.

In the field of very large-scale integrated technology, energy utilization has become a main challenging task for researchers due to the increase in the number of components on integrated circuits. Vedic mathematics is the oldest mathematical technique used to perform complex calculations using Vedic sutras in an easier and quicker approach. Designing of low power integrated circuit guided to the innovation of adiabatic logic. The combination of Vedic mathematics and adiabatic logic technique together gives high speed and power proficient multipliers. In this paper, the authors [7] proposed the design of an 8-bit Vedic multiplier based on a novel adiabatic logic known as corresponding pass transistor adiabatic logic. To validate the dominance of the projected work, the performance examination is compared with complementary metal-oxide semiconductor and proficient charge recovery logic. The output authenticates the decrease in propagation delay product of the corresponding pass transistor adiabatic logic Vedic multiplier.

In this digital era, there is a need for real time processing devices to achieve better outputs in real time applications. Such devices consist of arithmetic logic units and multiplier and accumulate units are used as a fundamental block for fast and effective computations. In digital signal processing applications to sustain the precision and high speed of computation, multipliers have to be advanced for which arithmetic logic unit and multiplier and accumulator unit be capable of providing better improvement. To increase the calculation speed of multiplication, there is an extremely high demand for implementing a faster multiplier. Vedic multipliers are mostly used for their speed of execution, region utilization and low power utilization. In the paper [8], out of sixteen Vedic sutras Urdhava Tiryakbhyam, Ekadhikena Purvena and Ekanyunena Pruvena are compared with simulation outputs. The main objective of the proposed method is to obtain outputs for the region, speed and power consumption constraints for each of the sutras.

Low power plays an important function in very large-scale integrated circuits. Due to the high demand for digital signal processors, power dissipation has become the most important parameter to be minimized. In digital signal processing applications, multiply and accumulate is the frequently utilized operations. In digital signal processors, its performance is decided by the parameters like power, look-up table utilization and propagation delay. The authors [9] have proposed architecture of 16-bit multiply and accumulate unit employing 8-bit Vedic multiplier and carry save adder. The authors compared the proposed multiplier with the traditional square root carry select adder array multiplier. The proposed Vedic multiplier architectural design attains considerable enhancement in region and delay.

In Vedic mathematics. the Vedic sutras provide the way or logic to compute complex mathematical operations which are extremely complicated to answer in the conventional technique. Intentionally or unintentionally human beings use Vedic sutras. For example, in programming, if we write j<--j-1 we use Ekanyunena Purvena sutra. The authors proposed and developed a graphical user interface window for Vedic addition and Vedic multiplication calculator implemented for Android based systems. Through the assistance of Vedic mathematics, the proposed application decreases the complication of mathematical computations. Vedic addition and multiplication technique validates the faster output than conventional techniques and it also utilized less power. [10]

Digital signal processing and communication applications mostly employed floating point multipliers for the computation of complex arithmetic operations. Therefore, there is increasing demand for the implementation of field programmable gate array-based structures. In this paper, the authors proposed a 32-bit floating point multiplier utilizing 4 floating point real multipliers resolution with the least path delay. For mantissa multiplication operations every floating-point multiplier needs a 16-bit fixed-point real multiplier. The authors proposed two structures for 24-bit Vedic real multiplier for mantissa multiplication in 16-bit floating point real multiplier. The proposed structures are assessed to select one structure having less propagation delay. The 4 floating points real multiplier solution is programmed in very high-speed integrated circuit hardware description language. [11]

In various digital signal processing and multimedia applications squaring and cubing of binary numbers plays an important role which incorporates least mean square, image compression, adaptive filtering, decoding and demodulation. The authors proposed novel squaring structures for binary numbers employing two basic Vedic sutras known as Dwandwa sutra and Antyayordashakepi sutra. Based on the proposed squaring structure and Anurupya sutra, a novel cubing structure is also proposed. Field programming gate technology has been employed for the implementation of 8-bit and 16-bit size squaring units and 8-bit sizes for cubing units. [12]

In digital signal processing and multimedia applications, multiply and accumulate operation is the most important operations extensively used. Multiplier is the essential element of digital signal processors. Its constraints like look-up table consumption, power and delay decides the evaluation of a digital signal processors. Therefore, it increases the demand to create a delay and power effective multiplier. The authors proposed a 16-bit multiplier and accumulate unit implemented using a Vedic multiplier and carry save adder. The proposed multiplier and accumulate unit are compared with the traditional array multiplier. The complete plan is realized in Verilog hardware description language. The proposed unit attains considerable enhancement in region and delay. [13]

For the enhancement of modified digital methodology, the most demanding task is to design devices that support superior, quicker and best possible digital computational structures. Based on Vedic sutras, the Vedic multiplier is considered as one of the promising computational algorithms that execute high speed computations easily. In the traditional Vedic multiplier field programmable gate arrays realization cannot carry optimization in terms of region, power utilization and delays. The authors [14] have proposed the implementation of a novel modern Vedic multiplier structure by employing the addition of two modules with dissimilar adder configurations for performance enhancement in terms of look-up tables, power utilization and delays. The proposed implementation structure efficiently attained a significant decrease in look-up tables, power utilization and delays when compared with the traditional Vedic multiplier structure.

Generally, in multiplication operation, the bits are multiplied bit by bit and then the multiplication results are summing up to get the final output. When the numbers of bits are raised for multiplication, the device turned out to be more complicated and the time consumes to generate output also raises greatly. Therefore, multiplication process needs a large quantity of hardware circuitry for handing out and the key objective during utilization is to achieve higher speed with low power utilization. Pipelining method is employed to decrease the energy utilization of the system. The authors proposed a novel system in which pipelining method was combined with Vedic mathematics to produces output with high speed and low power. The design algorithm of the proposed system programmed in VHDL realized o field programmable gate array. The proposed system achieves reduce in power utilization and produces the outputs faster. [15]

I. Fundamentals of Vedic Mathematics

Vedic mathematics comprise different techniques for execution of the complex mathematical functions through simplified processes. It provides simplified solution for the arithmetic, geometry, trigonometry, quadratic equations, factorization, and calculus. These simplified solutions are through different Vedic formulas. The description is through 16 main formulas and 13-sub formulas. It also provides solution for complex multiplication process. Out of the 5 different techniques of the multiplication as suggested through the Vedic mathematics, only one that is Urdhav Tiryagbhyam is the formulas which is generic in nature and can be applied to any condition of multiplication. Rest of the 4 formulas can be used if certain conditions are met. The different formulas which can be used for multiplication are listed through the table below.

TABLE I. Vedic Formulas For Multiplication

Sr. No.	Formula	Meaning	Technique
1.	Nikhilam Sutra	All from 9 and last from 10	Specific
2.	Anurupyena Sutra	Proportionately	Specific
3.	Ekayunena Purvena	By one less than the previous one	Specific
4.	Antyaordaske'pi	Last Totaling 10	Specific
5.	Urdhav Tiryagbhyam	Vertically and crosswise	**Generic**

Since the Urdhav Tiryagbhyam formula is generic and can be used for any condition of multiplication, it is preferred for architectural description of the proposed multiplier. The design of the proposed architecture is carried out and it is implemented using FPGA design flow and ASIC design flow. Both implementation flow and statistical outcomes are disclosed through the subsequent sections.

II. Implementation: Functional Verification

In this step, the known set of input values are applied to the component described architecture and output response is observed. If the output responses are closer to the expected one, the proposed architecture is concluded to be ok. Otherwise, the architecture is modified. Such verification, at the early stage, assures quick time to market. The outcome of the simulation process is discussed through the subsequent figures.

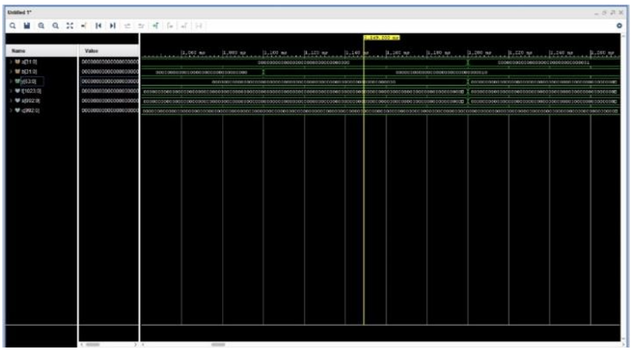

Fig. 1. Simulation View (a)

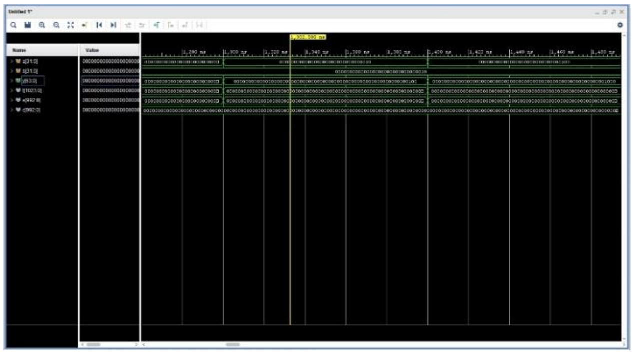

Fig. 2. Simulation View (b)

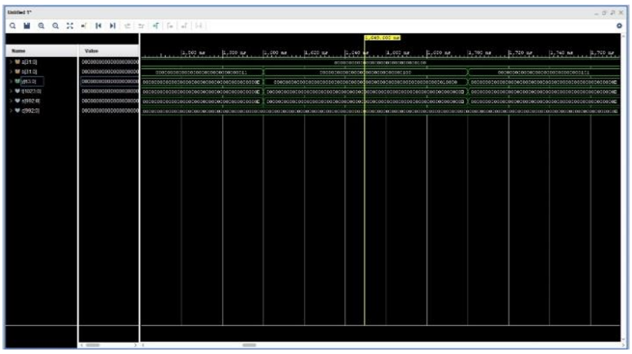

Fig. 3. Simulation View (c)

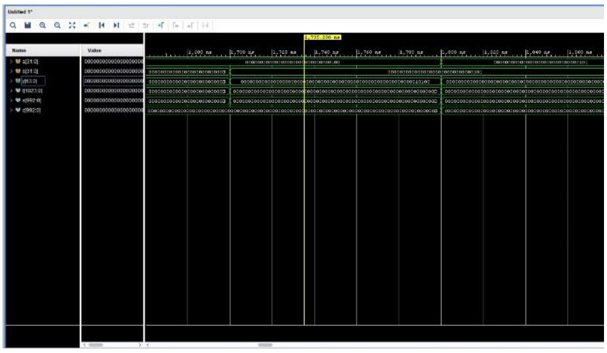

Fig. 4. Simulation View (d)

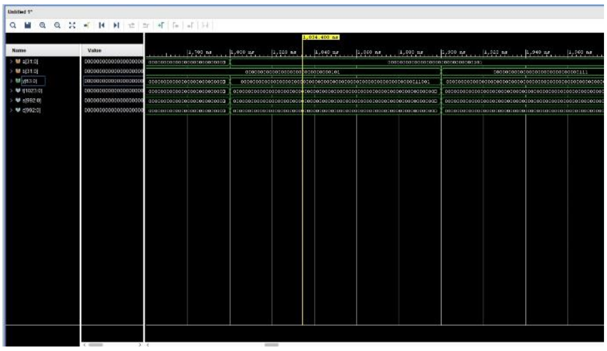

Fig. 5. Simulation View (e)

Implementation: Synthesis

Synthesis is the process in which the HDL statements are converted into the boolean statements and further to the actual hardware interconnect. Through the synthesis process RTL analysis is carried out which assures total conversion of the HDL statements into the workable hardware components. Some of the RTL schematic recorded are disclosed through the subsequent figures.

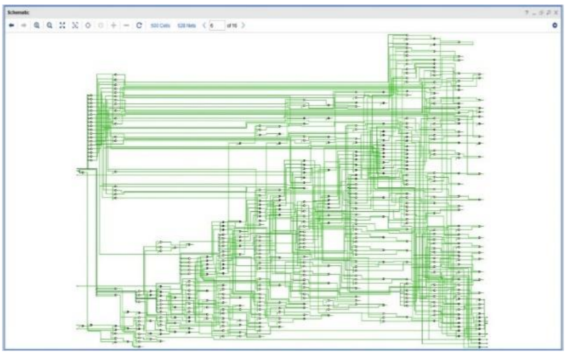

Fig. 6. RTL View (a)

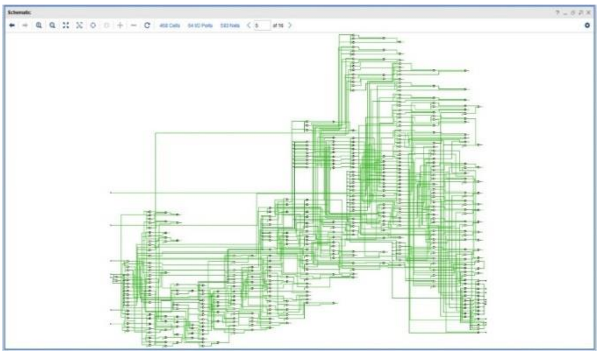

Fig. 7. RTL View (b)

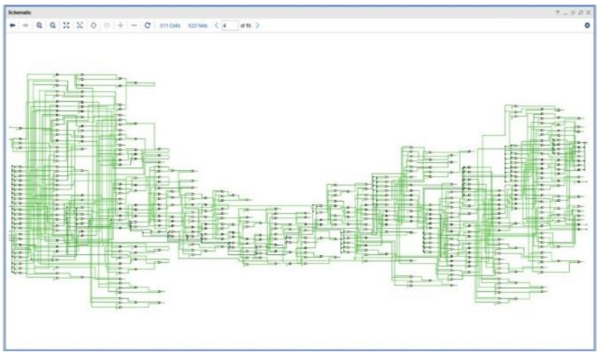

Fig. 8. RTL View (c)

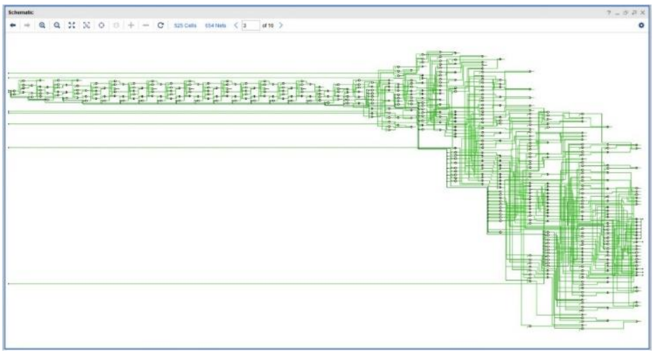

Fig. 9. RTL View (d)

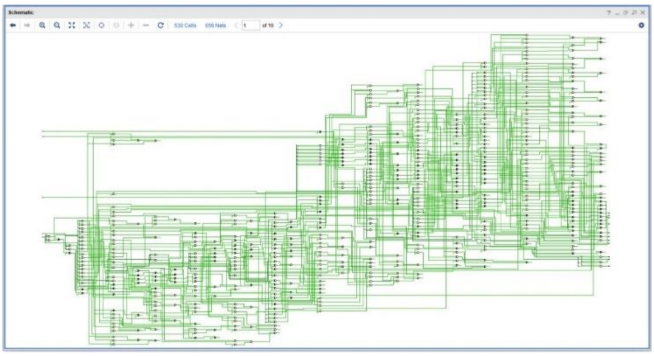

Fig. 10. RTL View (e)

Soon after the synthesis process, the implementation steps is carried out. This is actual mounting of the hardware created through the HDL synthesis process. In the implementation step different tiny steps like place, map and route are executed. In these steps, the hardware components are initially temporarily mapped, and level of optimization is observed by the tool. Once the balanced optimization with respected to the user constraints are met, the HDL converted hardware components are placed and then are interconnected through the routing process. It is the final stage of the FPGA design flow, after which the statistical analysis is carried out.

The outcome of the statistical analysis is disclosed, with respect to the time, frequency, power and resource utilization, are disclosed through the subsequent tables.

TABLE II. STATISTICAL ANALYSIS OF PROPOSED MULTIPLIER ARCHIETCTURE TARGETTING TO XILINX VIRTEX-7 FPGA

Parameters	V32	V16	V8	V4
Frequency Analysis (MHz)	387.597	465.983	474.608	486.145
Time Analysis (ns)	2.580	2.416	2.107	2.057
Power Analysis (W)	0.164042	0.047644	0.014422	0.004429
Resource Analysis (#)	2940	763	189	44

Considering the different ascendency parameters recorded through the above table, in the first impression it is evident that all of the parameters are up to the mark and can be considered as the highly optimized one. But when we consider the time parameter and detail statistics are recorded by splitting the time into delay contributed by the logic and delay contributed by the routing, amazing parameters have been seen. These are disclosed through the subsequent table.

TABLE III. TIMING ANALYSIS

Multiplier Size	Logic Delay (ns)	Routing Delay (ns)	Total Delay (ns)
V4	1.280	0.776	2.057
V8	1.341	0.776	2.107
V16	1.283	0.863	2.146
V32	1.285	1.295	2.580

It is observed through the above table that there is marginal difference of 0.5ns between the total delay contributed and the complexity is increased from 4-bit Vedic multiplier to the 32-bit Vedic multiplier architecture. The delay contributed by the logic components remains the same but on the other side of the coin, the routing delay is increasing significantly as the complexity is increasing and becomes more than that of the delay contributed by the logical components. This is because, the pre-compiled components are placed at different locations on the FPGA. Since the programmable architectures of the FPGA possess fixed hardware architecture which cannot be altered at any level of design. This has given a practical reason to move on to ASIC design for tight optimization.

Cadence is the versatile tool for implementation of ASIC flow. For completing the ASIC flow, Cadence nclaunch tool is used for simulation of the described architecture, Cadence Genus RTL Compiler is used for synthesis and Cadence Innovus RTL to GDS software tool is used for converting the netlist into gdsII file format for fabrication. The outcomes observed through tight optimization are disclosed through the subsequent table.

TABLE IV. TIGHT OPTIMIZATION WITH RESPECT TO TIME

Parameters	Technology Library	Time (ps)
Time Analysis (ps)	45nms	26248
	90nm	31079
	180nm	38715

The above table discloses that three different technologies viz. 45nm, 90nm and 180nm technology library are used for fabrication of the proposed architecture. While the timing analysis is carried out it is seen that the only 26248 ps of delay is contributed after smooth execution of the architecture which is the highest level of optimization for the proposed architecture using 45 nm technology library.

Conclusion

Targeting the hardware intensive architecture of the traditional multiplier component, an attempt is made to check the competency of the multiplier architecture designed using the fundamentals of the vedic mathematics. The effectiveness of the multiplier architecture designed using the vedic fundamentals are verified with respect to the different verticals like time, delay, frequency, resource utilization and power utilization. The assessment is carried out for 4-bit, 8-bit, 16-bit and 32-bit multiplier architecture.

The hardware architecture is described using the hardware description language. Post describing, the functional simulation is carried out to check the logical and functional correctness of the design. Once verified, the architecture is synthesized by targeting the design to the different variants of the FPGA. Immediately, the design is implemented in the FPGA in which various steps like Place, Map and Route functions are executed. While implementing the architecture in the FPGA, it is observed that the design is performing well with respect to the different parameters, but routing delay is way high as compared to the logic delay. In order to tightly optimize the mentioned parameters, the described architecture is explored through the ASIC design flow. In this flow, different tools of the cadence like Genus, Innovus and etc are used for optimization and implementation. Finally, tightly optimized results are seen when the design is implemented using the 45nm technology library file.

References

[1] M. N. Chandrashekara and S. Rohith, "Design of 8 Bit Vedic Multiplier Using Urdhva Tiryagbhyam Sutra With Modified Carry Save Adder," 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2019, pp. 116-120, doi: 10.1109/RTEICT46194.2019.9016965.

[2] K. Rajesh and G. U. Reddy, "FPGA Implementation of Multiplier-Accumulator Unit using Vedic multiplier and Reversible gates," 2019 Third International Conference on Inventive Systems and Control (ICISC), 2019, pp. 467-471, doi: 10.1109/ICISC44355.2019.9036345.

[3] A. Eshack and S. Krishnakumar, "Speed and Power Efficient Reversible Logic Based Vedic Multiplier," 2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC), 2019, pp. 1-5, doi: 10.1109/ICRAECC43874.2019.8995165.

[4] B. S. Krishna, P. V. Lakshmi and S. Musala, "Fault Resistant 8-Bit Vedic Multiplier Using Repairable Logic," 2019 International Conference on Emerging Trends in Science and Engineering (ICESE), 2019, pp. 1-5, doi: 10.1109/ICESE46178.2019.9194675.

[5] D. L. Prasanna and E. Prabhu, "An Efficient Fused Floating-Point Dot Product Unit Using Vedic Mathematics," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 12-15, doi: 10.1109/ICOEI.2019.8862718.

[6] A. Gupta, P. Kumawat and G. Kaur, "An Efficient Approach to Implement Multiplier using Vedic Maths Squaring Technique," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), 2019, pp. 90-93.

[7] K. Dutta, S. Chattopadhyay, V. Biswas and S. R. Ghatak, "Design of Power Efficient Vedic Multiplier using Adiabatic Logic," 2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2019, pp. 1-6, doi: 10.1109/UPCON47278.2019.8980057.

[8] S. Lad and V. S. Bendre, "Design and Comparison of Multiplier using Vedic Sutras," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), 2019, pp. 1-5, doi: 10.1109/ICCUBEA47591.2019.9128517.

[9] A. S. Krishna Vamsi and S. R. Ramesh, "An Efficient Design of 16 Bit MAC Unit using Vedic Mathematics," 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0319-0322, doi: 10.1109/ICCSP.2019.8697985.

[10] L. Mandal and K. Dasgupta, "Use of Vedic Mathematics to Speed-up Basic Mathematical Operations in Android Based Calculator," 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 162-165, doi: 10.1109/ISS1.2019.8907961.

[11] K. D. Rao, P. V. Muralikrishna and C. Gangadhar, "FPGA Implementation of 32 Bit Complex Floating Point Multiplier Using Vedic Real Multipliers with Minimum Path Delay,"

2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2018, pp. 1-6, doi: 10.1109/UPCON.2018.8597031.

[12] S. Barve, S. Raveendran, C. Korde, T. Panigrahi, Y. B. Nithin Kumar and M. H. Vasantha, "FPGA Implementation of Square and Cube Architecture Using Vedic Mathematics," 2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), 2018, pp. 6-10, doi: 10.1109/iSES.2018.00012.

[13] A. S. Krishna Vamsi and S. R. Ramesh, "An Efficient Design of 16 Bit MAC Unit using Vedic Mathematics," 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0319-0322, doi: 10.1109/ICCSP.2019.8697985.

[14] G. S. C. Teja, K. B. Sindhuri, N. U. Kumar and A. K. Vamsi, "Implementation of Vedic Multiplier Using Modified Architecture by Routing Rearrangement for High-Optimization," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), 2018, pp. 506-510, doi: 10.1109/CESYS.2018.8724037.

[15] A. Eshack and S. Krishnakumar, "Implementation of Pipelined Low Power Vedic Multiplier," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), 2018, pp. 171-174, doi: 10.1109/ICOEI.2018.8553853.