# DESIGN AND IMPLEMENTATION OF APPROXIMATE FLOATING-POINT VEDIC MULTIPLIER

**D. SWECHA,** Assistant professor, Electronics and Communication Engineering, Anubose institute of Technology, Palvancha, Telangana, India.
**S. SANDHYA,** Assistant professor, Electronics and Communication Engineering, Anubose institute of Technology, Palvancha, Telangana, India.
**V. RAVINDRA NAYAK,** Assistant professor, Electronics and Communication Engineering, Anubose institute of Technology, Palvancha, Telangana, India.
**K. VASAVI,** Assistant professor, Electronics and Communication Engineering, Anubose institute of Technology, Palvancha, Telangana, India.

## ABSTRACT

Now-a-days implementation of any design in digital electronics requires less hardware and software usage, less area, low power consumption, and less delay. Any idea implemented by satisfying any one of the above-mentioned requirements will give better performance of circuits. In most of the arithmetic operations, we will work on floating-point numbers which is a representation of real numbers as an approximation that increases the accuracy of our result. So floating-point computation is often found in systems that includes very small and very large real numbers, which require fast processing times.
As most of the applications need floating Point numbers multiplication. Floating point number is generally represented based on IEEE 754, having a sign, exponent, mantissa bits. So, our idea is to implement approximate floating-point multipliers using the approximate method. Because multipliers are key components of many high-performance systems, hence, optimizing the speed and power of the multiplier is a critical issue for an effective systems design. In the exact Floating-point multiplier is much like integer multiplication but it deals with unsigned integers and normalization. This technique is carried in three parts: first, we determine sign, next adding of exponent bits, and finally most important stage is finding the product of mantissa bits. This method requires high power, large storage and do not produce a finite number.
So, approximate/inexact computing has become an attractive approach for designing high-performance and low-power arithmetic circuits. The objective of this project is to implement an approximate floating-point multiplier with improving performance. For this, we make use of the Vedic multiplier which requires fewer gates, less area, and generates fast output. The Vedic multiplier is a popular multiplication scheme array that minimizes the number of adder stages required to perform the summation of partial products. This achieves relatively low area and the output can be obtained at a higher speed. In this project, the approximate floating-point architecture is simulated and synthesized by Xilinx (Viv ado tool). Hence an approximate computing is implemented and results, validations, comparisons are made, which makes a circuit effective.

## 1. Related work

According to Swagta Venkata Ramani [1] Recent years have witnessed significant interest in the area of approximate computing. Much of this interest stems from the quest for new sources of computing efficiency in the face of diminishing benefits from technology scaling. We argue that trends in computing workloads will greatly increase the opportunities for approximate computing, describe the vision and key principles that have guided our work in this area, and outline a range of approximate computing techniques that we have developed at all layers of the computing stack, spanning circuits, architecture, and software.
According to Langya Qian, Changhua Wang [2] Approximate/inexact computing has become an attractive approach for designing high performance and low power arithmetic circuits. Floating-point (FP) arithmetic is required in many applications, such as digital signal processing and machine learning. Different approximate FP multipliers are proposed in this paper, the accuracy and the circuit requirements of these designs are assessed to select the best approximate scheme as according to different metrics. It is shown that the proposed approximate single precision FP multiplier design

reduces power consumption, area and delay by up to 61%, 55%, and 49% respectively compared with its exact counterpart while incurring in a moderate error, moreover this paper shows that the so-called IFPM24-15 multiplier is the most efficient design in terms of PDP and NMED compared with previous inexact FP multipliers. High dynamic range (HDR) images are processed using the proposed approximate FP multipliers to show the validity of the approximate design.

According to Fabrizio Lombardi [3] Approximate circuits have been considered for error-tolerant applications that can tolerate some loss of accuracy with improved performance and energy efficiency. Multipliers are key arithmetic circuits in many such applications such as digital signal processing (DSP). In this paper, a novel approximate multiplier with a lower power consumption and a shorter critical path than traditional multipliers is proposed for high-performance DSP applications. This multiplier leverages a newly-designed approximate adder that limits its carry propagation to the nearest neighbors for fast partial product accumulation. Different levels of accuracy can be achieved through a configurable error recovery by using different numbers of most significant bits (MSBs) for error reduction. The approximate multiplier has a low mean error distance, i.e., most of the errors are not significant in magnitude. Compared to the Wallace multiplier, a 16-bit approximate multiplier implemented in a 28nm CMOS process shows a reduction in delay and power of 20% and up to 69%, respectively. It is shown that by utilizing an appropriate error recovery, the proposed approximate multiplier achieves similar processing accuracy as traditional exact multipliers but with significant improvements in power and performance.

According to Amir [4] Momani in exact (or approximate) computing is an attractive paradigm for digital processing at nanometric scales. Inexact computing is particularly interesting for computer arithmetic designs. This paper deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. These designs rely on different features of compression, such that imprecision in computation (as measured by the error rate and the so-called normalized error distance) can meet with respect to circuit-based figures of merit of a design (number of transistors, delay and power consumption). Four different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier. Extensive simulation results are provided and an application of the approximate multipliers to image processing is presented. The results show that the proposed designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design.

According to Bhardwaj [5] Today in sub-nanometer regime, chip/system designers add accuracy as a new constraint to optimize Latency-Power-Area (LPA) metrics. In this paper, we present a new power and area-efficient Approximate Wallace Tree Multiplier (AWTM) for error-tolerant applications. We propose a bit-width aware approximate multiplication algorithm for optimal design of our multiplier. We employ a carry-in prediction method to reduce the critical path. It is further augmented with hardware efficient precomputation of carry-in. We also optimize our multiplier design for latency, power and area using Wallace trees. Accuracy as well as LPA design metrics are used to evaluate our approximate multiplier designs of different bit-widths, i.e., 4 × 4, 8 × 8 and 16 × 16. The simulation results show that we obtain a mean accuracy of 99.85% to 99.965%. Single cycle implementation of AWTM gives almost 24% reduction in latency. We achieve significant reduction in power and area, i.e., up to 41.96% and 34.49% respectively that clearly demonstrates the merits of our proposed AWTM design. Finally, AWTM is used to perform a real time application on a benchmark image. We obtain up to 39% reduction in power and 30% reduction in area without any loss in image quality.

According to Kulkarni [6] We propose a novel multiplier architecture with tunable error characteristics, that leverages a modified inaccurate 2×2 building block. Our inaccurate multipliers achieve an average power saving of 31.78% - 45.4% over corresponding accurate multiplier designs, for an average error of 1.39% - 3.32%. Using image filtering and JPEG compression as sample applications we show that our architecture can achieve 2X - 8X better Signal-Noise-Ratio (SNR) for the same power savings when compared to recent voltage over-scaling-based power-error tradeoff methods. We project the multiplier power savings to bigger designs highlighting the fact that the

benefits are strongly design-dependent. We compare this circuit-centric approach to power-quality tradeoffs with a pure software adaptation approach for a JPEG example. We also enhance the design to allow for correct operation of the multiplier using a residual adder, for non-error-resilient applications

According to Anna Jain [7] Low-power is an imperative requirement for portable multimedia devices employing various signal processing algorithms and architectures. In most multimedia applications, the final output is interpreted by human senses, which are not perfect. This fact obviates the need to produce exactly correct numerical outputs. Previous research in this context exploits error-resiliency primarily through voltage over-scaling, utilizing algorithmic and architectural techniques to mitigate the resulting errors. In this paper, we propose logic complexity reduction as an alternative approach to take advantage of the relaxation of numerical accuracy. We demonstrate this concept by proposing various imprecise or approximate Full Adder (FA) cells with reduced complexity at the transistor level, and utilize them to design approximate multi-bit adders. In addition to the inherent reduction in switched capacitance, our techniques result in significantly shorter critical paths, enabling voltage scaling. We design architectures for video and image compression algorithms using the proposed approximate arithmetic units, and evaluate them to demonstrate the efficacy of our approach. Post-layout simulations indicate power savings of up to 60% and area savings of up to 37% with an insignificant loss in output quality, when compared to existing implementations.

According A. Gupta [8] design methodology for high-speed Booth encoded parallel multiplier. For partial product generation, we propose a new modified Booth encoding (MBE) scheme to improve the performance of traditional MBE schemes. For final addition, a new algorithm is developed to construct multiple-level conditional-sum adder (MLCSMA). The proposed algorithm can optimize final adder according to the given cell properties and input delay profile. Compared with a binary tree-based conditional-sum adder, the speed performance improvement is up to 25 percent. On average, the design developed herein reduces the total delay by 8 percent for parallel multiplier. The whole design has been verified by gate level simulation.

According to Hang Zhang [9] a low power probabilistic floating-point multiplier. Probabilistic computation has been shown to be a technique for achieving energy efficient designs. As best known to the authors, this is the first attempt to use probabilistic digital logic to attain low power in a floating-point multiplier. To validate the approach, probabilistic multiplications are introduced in a ray tracing algorithm used in computer graphics applications. It is then shown that energy savings of around 31% can be achieved in a ray tracing algorithm’s floating-point multiplier with negligible degradation in the perceptual quality of the generated image.

## 2. INTRODUCTION

**Motivation:**

Due to rapid changes in technology, designing a circuit with efficient methods will help to find a solution for complex problems. As the technology is developed a lot, there is no need of designing a new circuit, by modifying existed methods/results will give us an effective design that decreases our work and time and makes an easy process. Taking all these considerations implementing a digital circuit will benefit a lot in the future. Coming to the domain, as VLSI brings us the most change in our day-to-day life with nanotechnology makes us design a project in this domain.

**Problem Statement:**

In any arithmetic design, the main problem occurs while performing multiplication operations. As we already know that multiplier is the most power consumed design, so in any digital circuits play a key role so we need to design that consumes less power and area. And coming to our project a floating-point multiplier in the exact model requires more hardware and power due to multiplication. Initially, a multiplier consumes high power and floating- point consumes furthermore. As we already know multiplier problem but multiplier using floating-point is further more complex. On the other hand, floating point operations usually are slightly slower than integer operations, and you can lose precision.

But based on the operations we can control over the precision. As a result, they do not represent all of the same values, are not binary compatible, and have different associated error rates. Because of a lack of guarantees on the specifics of the underlying floating-point system, no assumptions can be made about either precision or range. By considering all these parameters we designed our project.

**3. Objective and Scope:**

The main objective of the project is to design an approximate floating-point multiplier with reduction in complexity. First, they can represent values between integers. Second, because of the scaling factor, they can represent a much greater range of values. The exact floating-point multiplier has more power consumption, area, the delay which is the drawback of the exact floating-point multiplier. So, we will design an approximate floating-point multiplier. To improve the power consumption, area, and delay. We, reduce some gates and adders. This can be accomplished by a Vedic multiplier which is an ancient multiplier but it has the accuracy of modifying to get the desired multiplier with the required elements. In Vedic one of the methods used for the approximate floating-point multiplier with uses low power, area, a delay is "Urdhva-Tiryagbhyam" (Vertically and Cross wise) Sutra with this sutra we can able to reduce some adders in the floating-point multiplier. Thereby decreasing power concerning to previous method i.e., the Exact floating-point multiplier will enable us to design a high-speed circuit that takes less power. Obtaining the constraints such as area, power, and delay help in determining the quality of the circuit used. If we need to estimate the quality of the logic circuit designed then we need to obtain the constraints of the circuit which are power, area, and delay. This is done in Xilinx Viv ado after the code is simulated. The results of the implementation of the generated Verilog HDL code give the required parameters. The obtained parameters are studied to analyze the designed model.

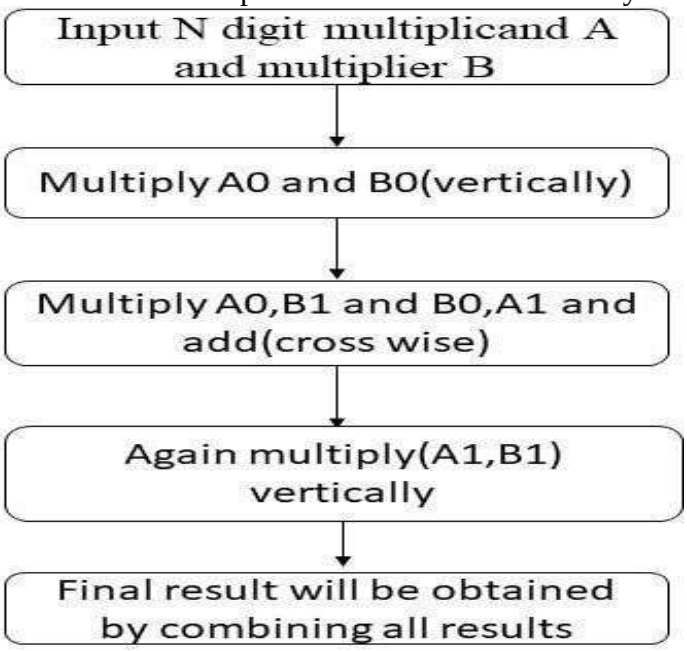



**Fig: Flow chart of Vedic multiplier based on UT sutra**

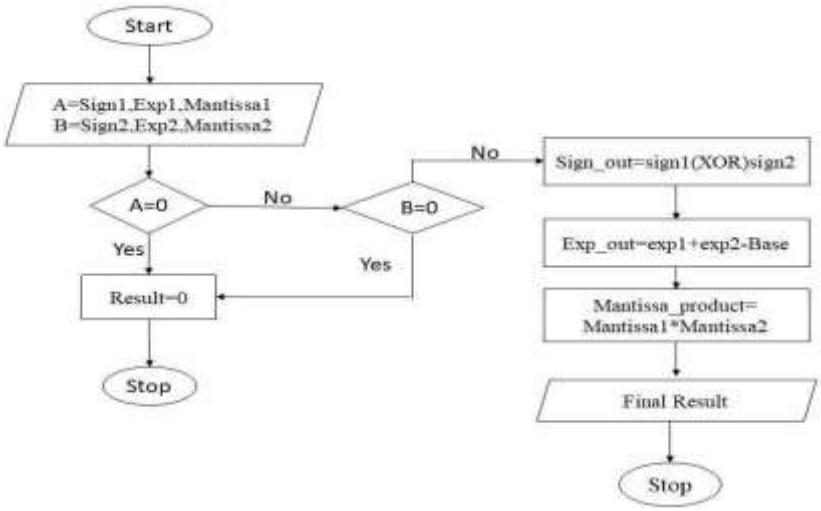


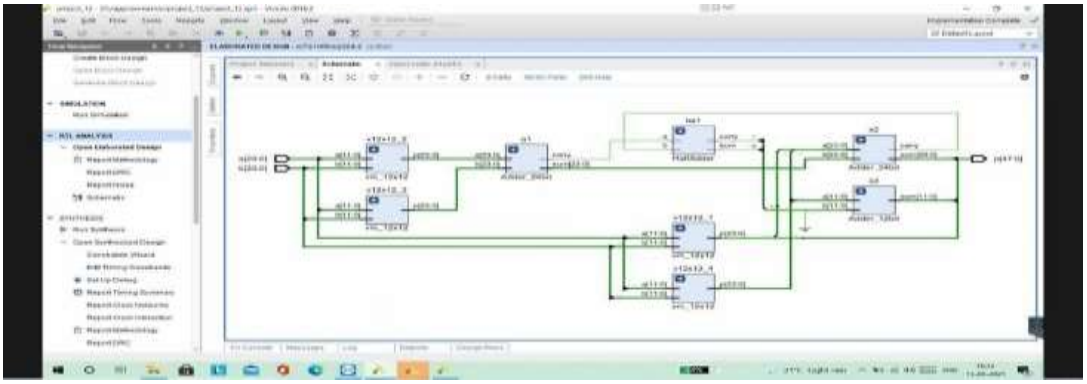

**Fig 4.1 Flowchart for exact FP multiplier**

## 4. RESULT

**Results of Exact model** code is developed in Verilog language and simulation results are obtained in Viv ado in both exact and approximate model. The results are as follows: In exact model the power and area consumed is more but the accuracy is hundred percent. The reason behind getting more power consumption is due to entire usage of the hardware to store the entire results. In present day scenario, the size of the on chip is decreasingday-by-day which evolves a new era in a nano technology world. So, we all ways need to takeusage of hardware or a space as a main constraint in VLSI technology. According to any designwe should concentrate on power, area in order to get efficient design for any high speed-basedcircuits. So, to decrease circuit complexity we go with approximate model.

**Figure below shows simulation for exact FP multiplier:**

**RTL schematic:**



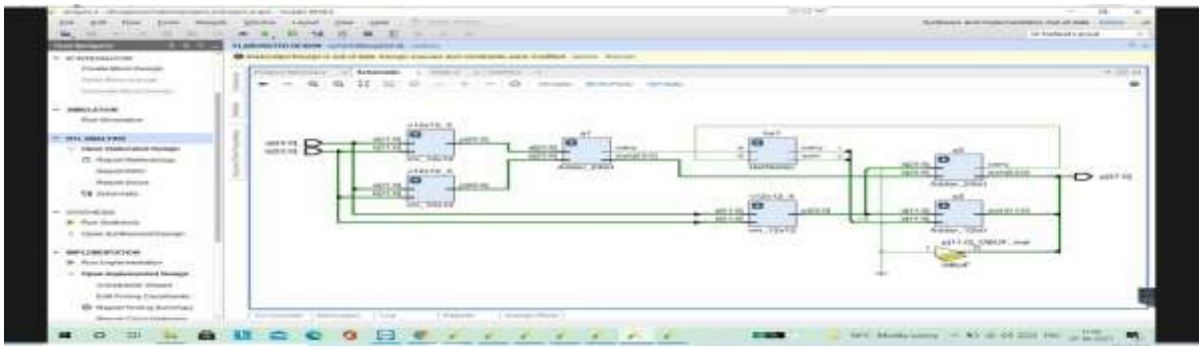
**Power:**

**Area:**

**Results of an Approximate model**

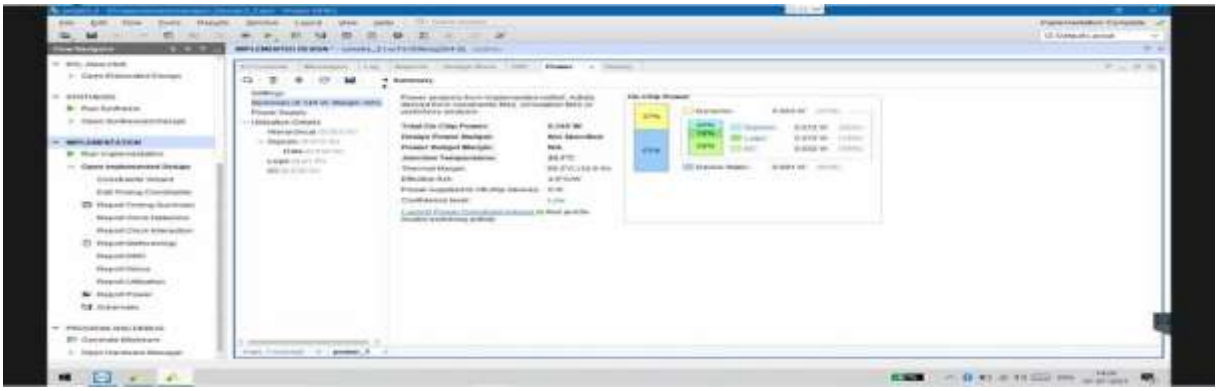
In approximate model the power and area are less compared to exact model because the circuit is reduced. Because we decreased the space constraint. The accuracy is varied based on the input we have given. For higher range the error is less.

**Figure below shows simulation for Approximate FP multiplier**

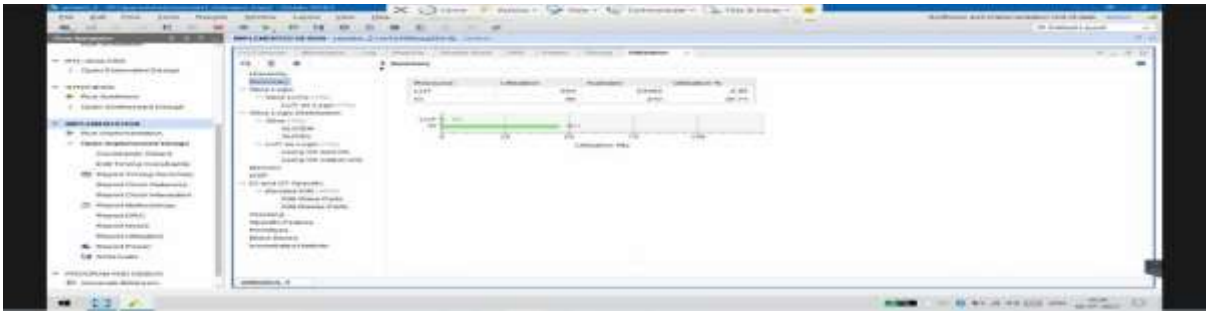


**RTL schematic**

**Power**



**Area**



**COMPARISON TABLES**

The scaling down of the transistor size lead to higher speed and higher integration density but increased power consumption density. The main losses in the CMOS circuits is dynamic losses in charging and discharging the parasitic capacitors of the transistors including the gate to source resistance. As the frequency increases the rate of charging and discharging increase leading to more power consumption. In fact, the power consumption is proportional to the frequency.

| S. No | Frequency | On-Chip-Power (Exact) | On-Chip-Power (Approximate) |
|---|---|---|---|
| 1. | 1KHz | 0.091W | 0.091W |
| 2. | 10KHz | 0.091W | 0.091W |
| 3. | 100KHz | 0.091W | 0.091W |
| 4. | 1MHz | 0.092W | 0.092W |
| 5. | 10MHz | 0.098W | 0.096W |
| 6. | 20MHz | 0.098W | 0.096W |
| 7. | 30MHz | 0.099W | 0.097W |
| 8. | 100MHz | 0.161W | 0.145W |
| 9. | 1GHz | 0.791W | 0.626W |

**Table 4.1** Power variation w.r.t Frequency

We have a relation between power and frequency given as:

$$P=f\ C_L V_{DD}^2$$

V= operating input voltage f(clk)= operating frequencyC= load

So, based on the above formula power is varied linearly with respect to the frequency clock. If frequency increases, active power generation increases, according to the ratio, the power factor also increases. This is the relation between frequency and power factor. To reduce power usage, clock frequency, reduction of switching activity, voltage scaling is very widely used. This technique is a very popular technique mainly used for the reduction of dynamic power dissipation. At 1GHz frequency,

the exact model is 0.791W and the approximate model is 0.626W that means 20% of power can be saved. The below table is about area and power at 100 MHz frequency.

| Parameter | Exact FP Vedic multiplier | Approximate FP Vedic multiplier |
|---|---|---|
| Power | 0.161W | 0.145W |
| Number of 4 input LUT'S | 775 out of 63400(1.2%) | 604 out of 63400(0.9%) |
| Number of slices | 223out of 1585(10.85%) | 172 out of 1585(10.85%) |
| Number of IOB's | 96 out of 210(45%) | 96 out of 210(45%) |

**Table 4.2** Power, Area of Exact and approximate model

| **Exact FPVedic multiplier** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Power** | | **Setup** | | **Hold** | |
| **S.No** | **Voltage** | **On-chip** | **Static Power** | **(logic delay) ns** | **PDP** | **(logic delay) ns** | **PDP** |
| 1. | 0.950 | 0.156 W | 0.089 W | 6.676 | 1.041 | 1.361 | 0.212 |
| 2. | 0.960 | 0.157 W | 0.089 W | 6.676 | 1.048 | 1.361 | 0.213 |
| 3. | 0.970 | 0.158 W | 0.090 W | 6.676 | 1.054 | 1.361 | 0.215 |
| 4. | 0.980 | 0.159 W | 0.090 W | 6.676 | 1.061 | 1.361 | 0.216 |
| 5. | 0.990 | 0.160 W | 0.091 W | 6.676 | 1.068 | 1.361 | 0.217 |
| 6. | 1.00 | 0.161 W | 0.091 W | 6.676 | 1.074 | 1.361 | 0.219 |
| 7. | 1.050 | 0.167 W | 0.094 W | 6.676 | 1.114 | 1.361 | 0.227 |

**Table 4.3** Power variation w.r.t Voltage (Exact)

| **ApproximateFP Vedic multiplier** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Power** | | **Setup** | | **Hold** | |
| **S.No** | **Voltage** | **On-chip** | **Static Power** | **(Logic delay)ns** | **PDP** | **(Logic delay) ns** | **PDP** |
| 1. | 0.950 | 0.140 W | 0.089 W | 5.484 | 0.767 | 1.359 | 0.190 |
| 2. | 0.960 | 0.141 W | 0.089 W | 5.484 | 0.773 | 1.359 | 0.191 |
| 3. | 0.970 | 0.142 W | 0.090 W | 5.484 | 0.778 | 1.359 | 0.192 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4. | 0.980 | 0.143 W | 0.090 W | 5.4 84 | 0.7 84 | 1.3 59 | 0.1 94 |
| 5. | 0.990 | 0.144 W | 0.091 W | 5.4 84 | 0.7 89 | 1.3 59 | 0.1 95 |
| 6. | 1.00 | 0.145 W | 0.091 W | 5.4 84 | 0.7 95 | 1.3 58 | 0.1 97 |
| 7. | 1.050 | 0.150 W | 0.094 W | 5.4 84 | 0.8 22 | 1.3 59 | 0.2 03 |

**Table 4.4** Power variation w.r.t Voltage (Approximate)

From tables 4.3 and 4.4, we will observe the variation of power w.r.t change in voltage. In the exact model, the on-chip power is more than in approx. the model even at the same voltages. As we know from the equation given above the voltage is directly proportional to power. So, an increase in voltage results in an increase in power. The power–delay product (PDP) is a figure of merit correlated with the energy efficiency of a logic gate or logic family. It is also known as switching energy, it is the product of power consumption i.e., on-chip power (averaged over a switching event) times the input- output delay or duration of the switching event (setup or hold). Because dynamic power dissipation depends on the square of the supply voltage and linearly on the frequency ($P = CV2 f$), if both the supply voltage and frequency are scaled-down, there is a cubic reduction in power consumption. When switching is done at high frequencies the overall power will be mostly due to dynamic power only.

**Comparison Parameters:**

**Dynamic power** refers to the component of the power dissipated in the CMOS circuit when the inputs switch from one level to another. It is the major component of the power dissipated in circuits and also contributes to peak power. This occurs only when there is switching activityso overall power is caused due to this dynamic power only

**Static power** is the power dissipation that occurs in the form of leakage power when the poweris not powered or is in standby mode. In circuits, there are several sources of leakage current including sub-threshold leakage, diode leakages around transistors and n-wells, gate leakage. This power is less because this amount of power is due to the inactivity of a circuit.

**On-Chip power:** It represents the total power consumed within the device. This can help determine the amount of power being consumed and dissipated by the device. The total on chip-power consumption per device is the sum of a dynamic component from charging and discharging the capacitance and a static component from the leakage current.

## 5. CONCLUSION

In this project, multiplier plays a key role. In the Exact FP multiplier, each operation requires a greater number of adders, Multipliers, shifters. So, it requires a greater number of storages. So, it consumes more power and the result will be delayed and we are having high hardware requirements along with complexity in design. For this, there will be more work is need to be done and do not produce a finite value. So that we have taken the approximation model to lower the power. So that we have approximated mantissa part. So Vedic multiplier isused for mantissa multiplication. In the Vedic we have used the sutra called Urdhva - Tiryagbhayam which is used for reducing delay power and hardware requirement.

Coming to the coding we have used the Verilog because the Verilog it is having a smaller number of codes compared to VHDL and it is easy to understand. And at last, we havetaken the comparison for the Exact and approximate FP multipliers. Hence the normalmultiplication method needs more power, area, and hardware So, using the approximation of multiplication with Vedic multiplier reduces the power, area parameters. The most important matter is that performance speed is increased through this approach.

**References**

[1]. Swagath Venkataramani, Srimat T. Chakradhar, Kaushik Roy and Anand Raghunathan. "Computing Approximately, and Efficiently," Design, Automation& Test in Europe Conference & Exhibition, pp.748-751, 2015.

[2]. Liangyu Qian, Chenghua Wang, Weiqiang Liu, Fabrizio Lombardi, and Jie Han, "Design of Approximate Wallace-Booth Multipliers," IEEE International Symposium on Circuits and Systems (ISCAS), May 2016. (accepted).

[3]. Cong Liu, Jie Han, Fabrizio Lombardi. "A Low-Power, High Performance Approximate Multiplier with Configurable Partial Error Recovery," Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 1-4, 2014

[4]. Amir Momeni, Jie Han, Paolo Montuschi, Fabrizio Lombardi. "Design and analysis of approximate compressor for multiplication," IEEE Trans. Computers,vol. 64, no. 4, pp. 984-994, 2015.

[5]. BhardwajK., Mane P.S., Henkel J. "Power-and area-efficient Approximate Wallace Tree Multiplier for error-resilient systems," 15th International Symposium on Quality Electronic Design (ISQED), pp.263-269, 2014.

[6]. KulkarnP., GuptaP., ErcegovacM., "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," 24th International Conference on VLSIDesign (VLSI Design), pp.346-351, 2011.

[7].Anna Jain;Baisakhy Dash;Ajit Kumar Panda;Muchharla Suresh2012 International Conference on Devices, Circuits and Systems (ICDCS)

[8]. A. Gupta, S. Mandavalli, V. Mooney, K. Ling, A. Basu, H. Johan, and B. Tandianus, "Low power probabilistic floating point multiplier design," Proc. IEEE Comput. Soc. Annu. Symp.VLSI, pp. 182-187, 2011.

[9]. Aman Gupta, Hang Zhang, Wei Zhang, and Lach, J., Low power probabilistic floating point multiplier design, Low power probabilistic floating point multiplier design,32nd IEEE International Conference onComputer Design (ICCD), pp. 48-54, 20.