1. **Introduction**

In the landscape of major projects in digital circuit design, the quest for efficient and high-performance arithmetic units remains a critical pursuit. One such area of exploration is the design and implementation of an 8-bit Vedic multiplier, leveraging compressor-based techniques and harnessing the benefits of FinFET (Fin Field-Effect Transistor) technology. This introduction sets the stage for a comprehensive exploration of this innovative approach, highlighting the project's relevance and objectives.

The multiplication operation serves as a fundamental building block in numerous digital applications. Conventional multiplication approaches often exhibit inherent complexities, resulting in increased power consumption and prolonged critical paths. The utilization of Vedic mathematics principles offers a paradigm shift, providing a modular and parallel approach to multiplication, making it an intriguing choice for major projects demanding high-performance arithmetic units.

Central to this major project is the incorporation of compressor architectures within the Vedic multiplier. Compressors play a pivotal role in optimizing the processing of partial products by simultaneously handling multiple bits, thereby reducing critical path delays and overall power consumption. This integration aims to elevate the performance of the Vedic multiplier, rendering it suitable for major applications necessitating swift and energy-efficient arithmetic operations.

The project strategically adopts FinFET technology, a cutting-edge advancement in transistor design. The three-dimensional fin-like structure of FinFET addresses the limitations of traditional planar transistors, providing enhanced control over the channel, mitigating short-channel effects, and improving overall transistor characteristics. This integration facilitates reduced power dissipation and increased scalability, aligning perfectly with the major project's goal of achieving high-performance arithmetic units.

The various steps comprise of addition, subtraction, and many comparisons which as a result consume much time therefore reduce the speed of multiplier. Since speed is major concern for Multiplier Design, therefore these architectures are not feasible. With a view to mitigating the aforesaid problems, we put forward a new technique to design multiplier with concept of Vedic Mathematics. FinFET (Fin Field-Effect Transistor) technology is a type of transistor design that offers several advantages over traditional CMOS (Complementary Metal-Oxide-Semiconductor) technology, making it a preferred choice for advanced semiconductor manufacturing processes.

* 1. **Overview**

This project, which revolves around the 8-Bit Vedic multiplier using compressor in FinFET technology, stands as an innovative exploration into the realm of arithmetic unit design. The introduction lays the groundwork by acknowledging the challenges posed by traditional multiplication techniques and introducing the motivation behind incorporating Vedic mathematics principles. The primary objectives are explicit: design, implement, and analyze an 8-bit Vedic multiplier with compressor-based techniques within the framework of FinFET technology. The comprehensive literature review positions the project strategically to leverage the advantages of Vedic multiplication and compressor architectures while addressing challenges through the adoption of FinFET technology.

In the subsequent sections, the project follows a systematic approach with a focus on specific steps. The architectural intricacies of the 8-bit Vedic multiplier are explored, emphasizing its modular and parallel nature, and the integration of compressor-based techniques to optimize performance. Strategic implementation of FinFET technology is undertaken to enhance transistor characteristics, providing improved control over the channel and mitigating short-channel effects. The methodology section outlines a step-by-step process, including detailed design and the incorporation of compressor-based techniques. Simulation tools are then selected for a thorough analysis of performance metrics, incorporating previous results to showcase the evolution of the project.

* 1. **Objective**

The main objective of developing an 8-bit Vedic multiplier using compressor techniques in FinFET technology is to create an efficient and high-performance arithmetic unit for digital circuits. This project aims to leverage the advantages of Vedic mathematics principles, compressor architectures, and FinFET technology to address key challenges in conventional multiplication approaches.

Efficiency enhancement, technology integration, speed improvement, power efficiency, validation through simulation, contribution to digital circuit design are some of the things that are primary goals of this project. Utilizing the Cadence software, a widely-used electronic design automation tool, for the design, simulation, and verification of the 8-bit Vedic multiplier. Cadence tools, such as Virtuoso, Encounter, or other relevant modules, will play a crucial role in the design flow, from schematic capture to layout and simulation.

Conducting the comprehensive performance analysis using Cadence tools to evaluate key metrics such as speed, power efficiency, and area utilization is done. This includes simulations to validate the functionality and performance of the multiplier under various conditions.

Performing a comparative study with conventional multipliers will highlight the advantages of the proposed 8-bit Vedic multiplier. The comparative study between an 8-bit Vedic multiplier using compressor architecture in FinFET technology and conventional 8-bit multipliers employing adders in CMOS technology involves a detailed analysis across various performance metrics.

Evaluating the efficiency gains achieved by the proposed Vedic multiplier in comparison to CMOS-based multipliers. Focus is on computational speed as a primary metric, aiming to showcase reduced critical path delays and improved overall speed. Analyzing the power consumption characteristics of each design and comparing dynamic power consumption during active computation and static power consumption when the circuit is idle. Assessing the physical size of each design by comparing area utilization, evaluate whether the Vedic multiplier, with its compressor architecture, demonstrates a more compact layout compared to CMOS-based multipliers using adders.

* 1. **Motivation**

The motivation for developing an 8-bit Vedic multiplier using compressor architecture in FinFET technology stems from a combination of historical mathematical principles, contemporary computational demands, and the advancements offered by state-of-the-art semiconductor technologies. Several key factors drive the motivation for this project:

* + - Efficiency and Speed: The Vedic multiplier, inspired by ancient Indian mathematical techniques, inherently offers parallelism and modularity. By leveraging these characteristics and integrating a compressor architecture, the aim is to improve the efficiency of multiplication operations. The compressor's ability to process multiple bits simultaneously is expected to reduce critical path delays, leading to increased computational speed.
    - Relevance of Vedic Mathematics: Vedic mathematics, originating from ancient Indian texts, provides alternative and efficient techniques for arithmetic operations. The Vedic multiplier, based on these principles, offers a unique perspective for enhancing computational efficiency. The motivation lies in exploring the applicability of Vedic mathematics to modern digital circuit design.
    - Parallel Processing Advantages: The Vedic multiplier divides the multiplication process into smaller, independent tasks, allowing for parallel processing of partial products. This approach aligns well with the parallel capabilities of modern processors, promising improvements in speed and overall efficiency. The project is motivated by the desire to harness the benefits of parallelism for 8-bit multiplication.
    - Integration of Compressor Architecture: The introduction of a compressor architecture further enhances the capabilities of the Vedic multiplier. Compressors efficiently process partial products, potentially reducing power consumption and critical path delays. The motivation lies in exploring how this integration can amplify the advantages of the Vedic multiplier in the context of 8-bit arithmetic operations.
    - Advancements in FinFET Technology: FinFET technology represents a significant advancement in semiconductor design. The three-dimensional fin-like structure of FinFETs provides better control over the channel, mitigates short-channel effects, and improves overall transistor characteristics. The motivation is to capitalize on these advantages to achieve a high-performance multiplier with reduced power dissipation.
    - Energy-Efficient Design: In the era of energy-efficient computing, the motivation for this project is to contribute to the development of arithmetic units that not only perform computations swiftly but also optimize power consumption. The combination of Vedic mathematics principles, compressor architecture, and FinFET technology aligns with the broader goal of energy-efficient digital circuits.
    - Potential Applications: The motivation extends to the potential applications of the 8-bit Vedic multiplier in various domains, including digital signal processing, image processing, and cryptography. Efficient arithmetic units are fundamental to the performance of these applications, making the exploration of innovative multiplier designs particularly relevant.
  1. **Methodology**

In this project we incorporate a compressor-based technique to increase the speed of Vedic mathematics-based Multiplier by changing Full adder and Half adder and using Compressor. Compressor can add more than three bits at a time and computes this operation with fewer numbers of gates and has higher speed as compared to full adder. Since the compressor requires few gates so area of compressor reduces in comparison to equivalent full. The flow of project is as show in figure 1.

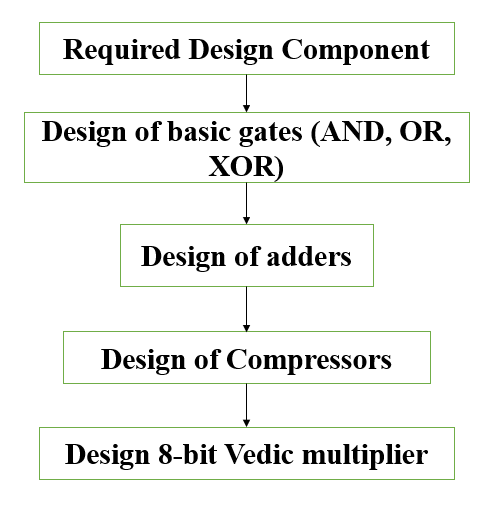


Fig 1.1: Design flow of proposed methodology

In this project we introduce an architecture of Vedic multiplier by using 4:2 compressors and 7:2 compressors for addition that increase the speed of Multiplier. The 7:2 compressors are made of 5:2 compressors and two full adders. In this project, a compressor based Urdhva-Tiryakbhyam architecture in is proposed which is based on the Urdhva-Tiryakbhyam Sutra of Vedic mathematics so as to enhance the speed of the binary multiplication.

The idea of designing the multiplier from Vedic mathematics has emerged as in this technique the partial products and the sums are generated in only a single step. These partial products are then added based on the Vedic math’s algorithm to obtain the final product. This in turn leads to a very high-speed approach to perform multiplication.

1. **Literature survey**

"According to Raj Kumari and Rajesh Mehra [1] Since multipliers are among the most crucial components of practically all digital system hardware, effective digital system designs will emerge from multiplier designs with high speeds, short delays, small areas, and low power consumption. Thus, using our traditional Vedic mathematical technique, the Urdhva Tiryakbhyam Sutra, this work presents an efficient design and construction of a multiplier with fast speed, decreased delays, minimal area, and low power consumption Utilizing an antiquated Vedic technique, the efficient Vedic multiplier is built in the 45nm VLSI Cadence Virtuoso tool. The implementation of the multiplier makes advantage of an efficient low power and high-performance adder circuit, allowing for the execution of a more effective Vedic multiplier design. Due to its carry skip addition mechanism, less hardware, and shorter delays, this multiplier is substantially faster than other conventional multipliers while also consuming less power Multiplier circuit delay is kept to0.2087us. This circuit's computed total power is a very small0.2854 mw. As a result, it is possible to deploy higher order multipliers in the future with extremely little power consumption and little delays.

According to the Chiranjit R Patel [2], Multiplier modules that are fast and space-efficient have been created that can be used in FIR filters and digital signal processors. Higher bit multipliers can be achieved for upcoming work and compared to conventional multipliers. Even when there is a large range of temperature fluctuation, multipliers based on this method have fast switching rates. Higher bit multipliers are possible, and they operate at a high frequency. As a result, it is feasible to implement high-speed DSPs and ASICs. The Urdhva Tiryakbhyam sutra to present a 4x4 multiplier with Carry Save Adder. Further, voltage ranges from 0.9V to 1.1V are used for the performance investigation. In the circumstances of greatest limitation, a delay of 250ps and a power usage of 3.795uw were obtained.

According to the S. Karunakaran [3], The mathematical formula Urdhva Tiryakbhyam is generic and can be used in all multiplication situations. Additionally, it has been noted that the architecture based on this sutra is comparable to the well-known array multiplier, where an array of adders is needed to produce the final result. Because of its structure, it experiences a significant carry propagation delay when multiplying large numbers. The Urdhva Triyakbhyam Sutra, which transforms the multiplication of two huge numbers into the multiplication of two little numbers, can resolve this issue. In order to develop high performance multipliers in VLSI applications, the potential of Vedic mathematics might be explored. Vedic Mathematics' Urdhva Triyakbhyam Sutra is less difficult than other sutras and may be tested by applying different logics to it in VLSI. When compared to Wallace tree multiplier and array multiplication, the Vedic multiplier uses 14.22% and 18.86% less power, respectively.

According to the S. Prema, [4] in a reversible gate, n inputs and n outputs are used. One to one mapping is the name of the format. The inputs may readily be used to determine the outputs, and the outputs can likewise be used to determine the inputs. Direct fan out is not permitted during in the fabrication of reversible circuits since one-to many concepts are not reversible. By adding more gates, we can somehow obtain the spread out in the reversible circuits. It's important to use the fewest possible reversible logic gates when developing reversible circuits. A variety of factors affect the circuit's performance and complexity. The two 2\*2 Reversible VMs are combined to create a 4\*4 Reversible VM. Two BME gates and two PERES gates are used to create a reversible 2\*2 multiplier. The total p0, which is the result of multiplying the LSB x0 and y0, is produced by the first BME gate. Two trash outputs are produced by each BME gate. The PERES gate receives one of the BME gate's outputs, which in turn generates the higher order bits for the sums s1, s2, and s3. The design is obtained 0.281mW power and 5353 ps of delay.

According to the Malti Bansal [5], it is possible to implement more sophisticated circuitry using multiplier circuits, including signal processors like convolutes, and then employ GDI method to further optimize those complex circuits. Here, a thorough comparison of the GDI and CMOS multipliers has been carried out utilizing FinFETs (18nm) for several characteristics, including transistor count, power, propagation delay, and power-delay product. It has been noted that the GDI multiplier outperformed traditional ones due to its sharp performance characteristics, which include a 31% reduction in propagation latency, a 55.6% reduction in power consumption, and a 35.12% reduction in transistor count. Given that multipliers are the most fundamental yet essential component of current ALUs, they can be further optimized utilizing the GDI technique to enhance their performance in various dimensions.

According to the Nehru Kandasamy [6], the 18 transistors (18T) full swing GDI based digital logic and transmission-based methodologies used in the high-speed adder circuit design compute the sum and carry bits, respectively. Operating at duty cycles of 1.8 and 1.2 V is the suggested adder logic, which is implemented in 180nm and 90nm technologies. In comparison to typical 16 transistor-based adder cells, the suggested design has less power delay effects and reduces 28.57%. The number of transistors in the suggested adder circuit is multiplied by 2. To develop 4-bit ripple carry, carry save, and carry select adder circuits with decreased electrical characteristics in regards to power, delay, and power delay products, the effectiveness of the proposed 1-bit adder circuit is demonstrated. In comparison to conventional methods, the findings of computational modelling have demonstrated higher circuit performance in terms of power-delay products. When using 180 nm technology, the proposed topology demonstrated power-delay product improvements of 49.646% for the presented Barun multiplier-based MAC unit and 30.84% for the presented Baugh Wooley multiplier-based MAC unit, while in 90 nm technology, the proposed circuit demonstrated power-delay product improvements of 52.09% for the presented Barun multiplier-based MAC unit and 11.71% for the presented Baugh Wooley multiplier-based MAC unit.

1. **Schematic diagram and Working principle**
   1. **Design of functional block**

In this section the design of functional blocks such as AND, OR and XOR gates, 2×2 vedic multiplier, 4-bit ripple carry adder, 4×4 vedic multiplier, 8-bit ripple carry adder are discussed in this section. All circuits are designed in cadence virtuoso.

* + 1. **AND gate**

A logic circuit containing two or much more inputs and one output is identified as an AND gate. The logic of logical multiplication is used by an AND gate. In this gate, the output is low (zero) when one of the sources is low. The output will be high(one) if all of the sources are high. The AND gate is designed in two ways, one was using traditional AND circuitry which uses 6 transistors to implement the logic as illustrated in Figure 3.1.

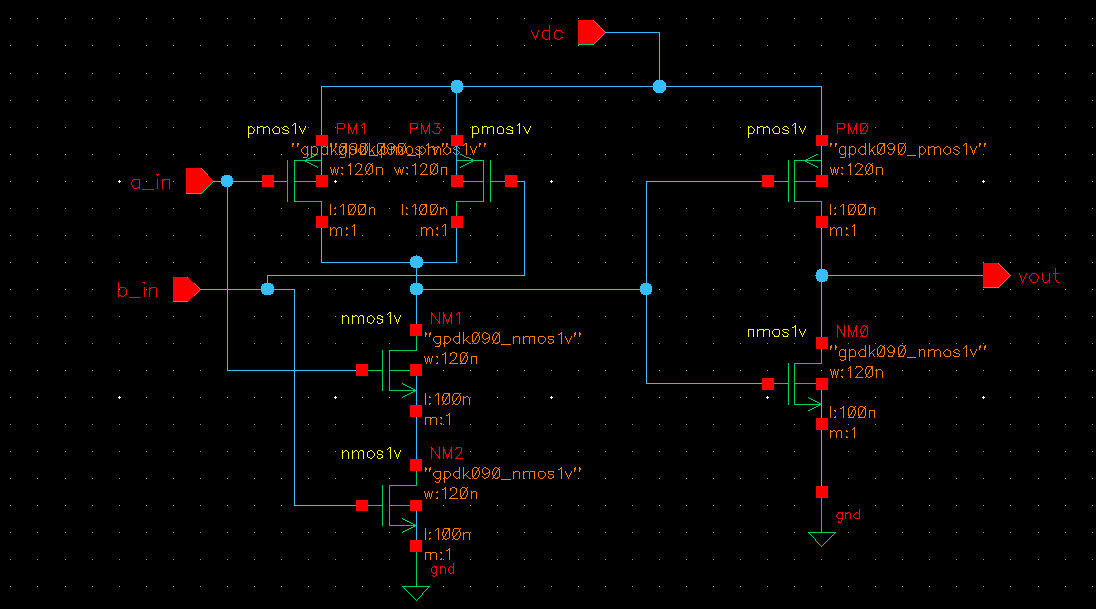


Fig 3.1: Schematic diagram of AND circuit

* + 1. **OR gate**

An OR gate, short for "logical OR gate," is a fundamental digital logic gate that performs a specific logical operation on one or more binary inputs to produce a single binary output. The OR gate outputs a high (logic 1) if at least one of its inputs is high, and it outputs a low (logic 0) only if all of its inputs are low. The schematic diagram of or gate is shown in figure 3.2.

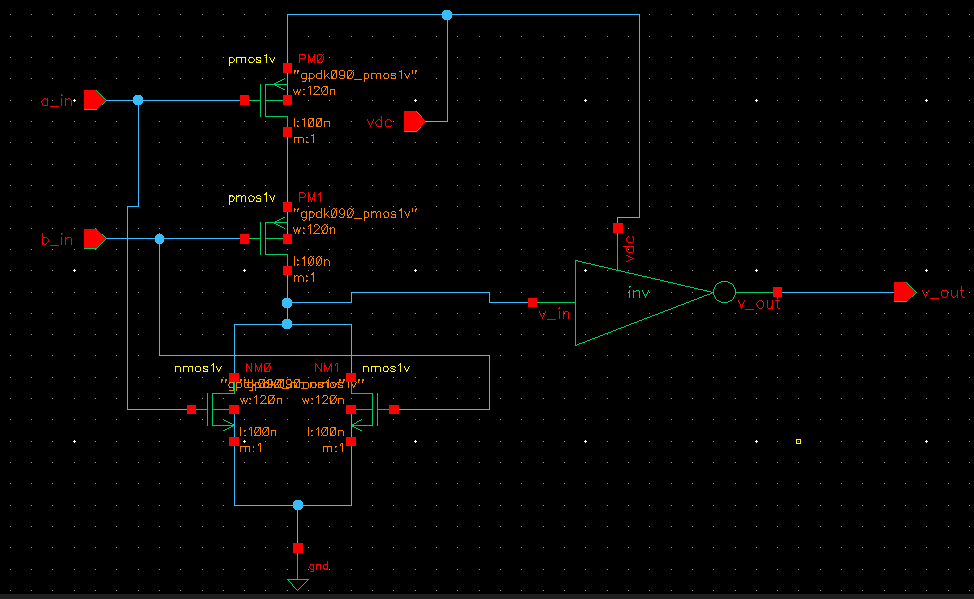


Fig 3.2: Schematic diagram of OR circuit

* + 1. **XOR gate**

Combining several common logic gates results in the XOR gate. The XOR gate is a common component in arithmetic logic circuits, computational logic comparators, and error detection circuits. When one of its two sources is "high" ("one") and the other is "low" ("zero"), the Exclusive OR gate simply provides an output ("one"). The XOR gate is having 12 transistors, as shown in Figure.3.3

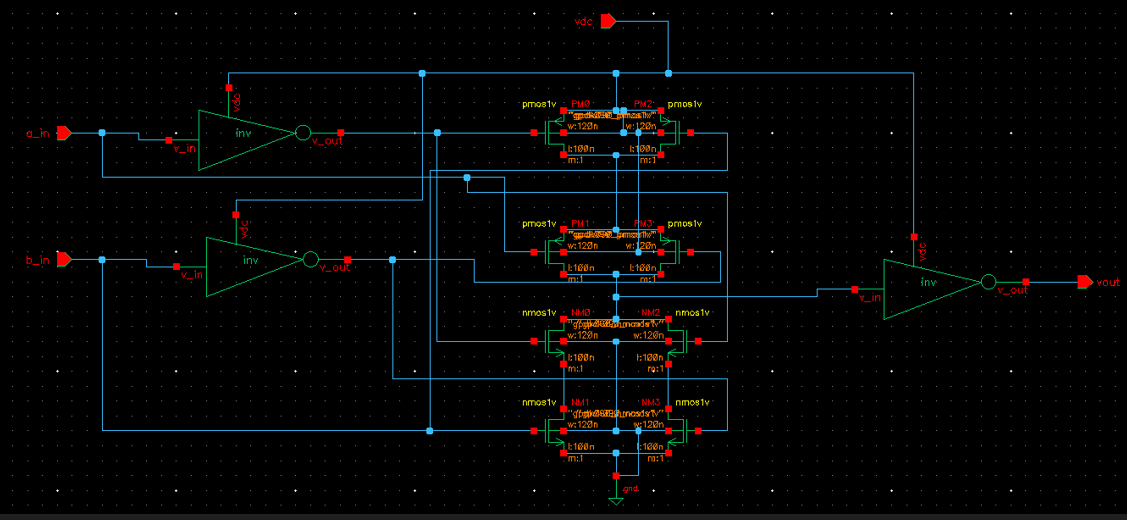
****

Fig 3.3: Schematic diagram of XOR circuit

* + 1. **Full adder**

The conventional full adder may be created using half adders, which are made up of AND and XOR gates. The two variables may be combined to create the sum expression (sum=A XOR B), and the carry expression (carry=A AND B). The full adder may be created using two half adders and one OR gate as described in the Figure 3.4.

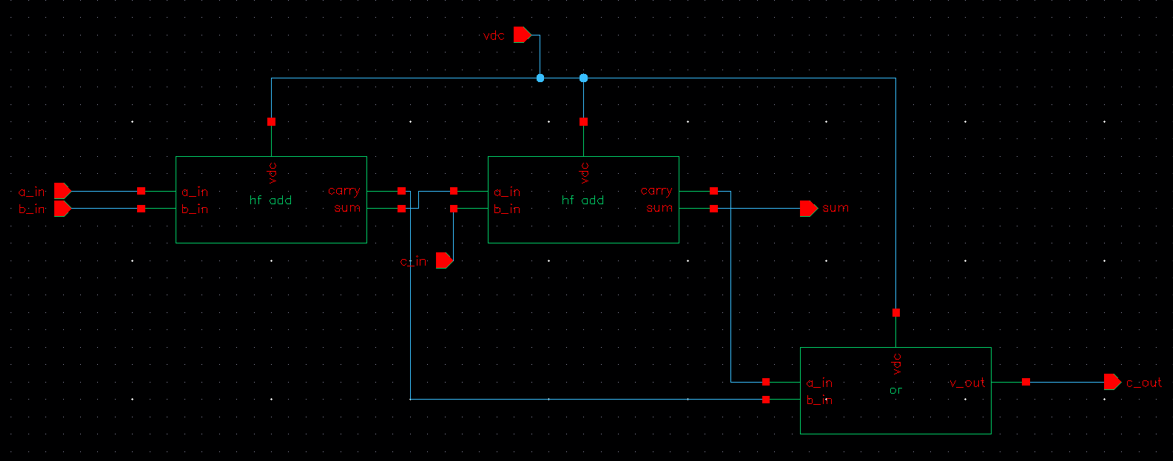


Fig 3.4: Schematic diagram of full adder

* + 1. **2×2 Vedic multiplier**

Consider the two 2-bit values A and B as shown in the "Fig.3.5", where A = "al a0" and B = "bl b0", The LSBs of A and B, "a0" and "b0", are multiplied at the start, creates the LSB("po") of the final product (vertical). The product of the LSB of the multiplier ("b0") and the MSB bit of the multiplicand ("al") and added to the product of the LSB of the multiplicand ("a") and the MSB bit of the multiplier ("b1") (crosswise). The sum returns the final product's second bit, "pl" and the carry value "cl" is added to the result which obtained by multiplying the MSB ("al" and "bl"), providing the sum and carry. The sum is the final product's third bit corresponds to "p2", whereas the carry is the fourth bit, "p3" as shown in Figure 3.5.

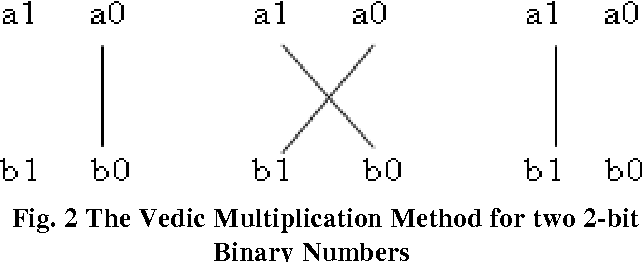


Fig 3.5: 2-bit vedic multiplication

The following is an example of a typical operation:

p0=a0b0 (i)

c1p1 = a1b0 + a0b1 (ii)

p3p2 = c1 + a1b1 (iii)

p3 p2 pl p0 will be the final result.

The conventional multiplier would conduct the same function as the traditional multiplier, multiplying the multiplicand's LSB and MSB with the multiplier and merging the results. As a result, there is no obvious improvement in the efficiency of the Vedic multiplier.

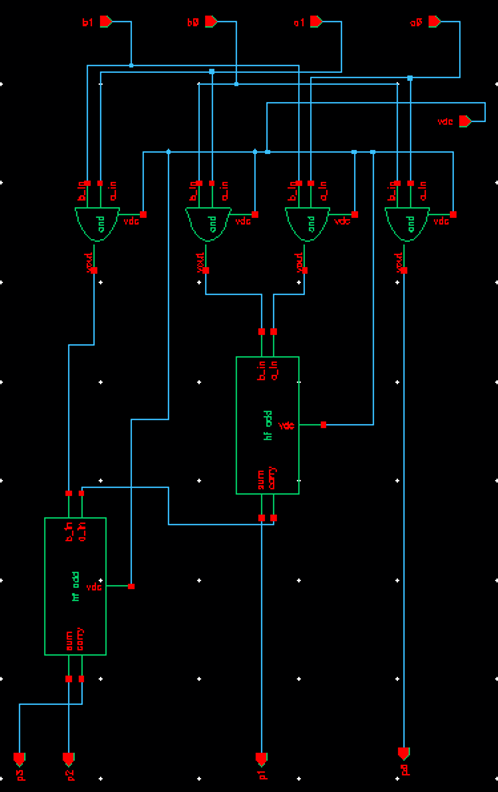


Fig 3.6: Schematic of 2×2 Vedic multiplier

The Urdhva Tiryagbhyam sutra performs 2x2 multiplication in a single line as shown in Figure 3.5, whereas the shift and add (conventional) method requires four partial products to be added to obtain the result. Thus, using the Urdhva Tiryagbhyam Sutra in binary multiplication reduces the number of steps required to calculate the final product, resulting in less computational time and faster multiplier speed. The schematic is shown in figure 3.6.

* + 1. **4-bit ripple carry adder**

The 4-bit RCA is implemented by using the full adder from above three different technology which one have better performance compare to others. Basically, RCA produce lesser delay other adder like "Carry Save Adder", "Carry Skip Adder", "Carry Lookahead Adder", etc... The schematic is as shown in Figure 3.7

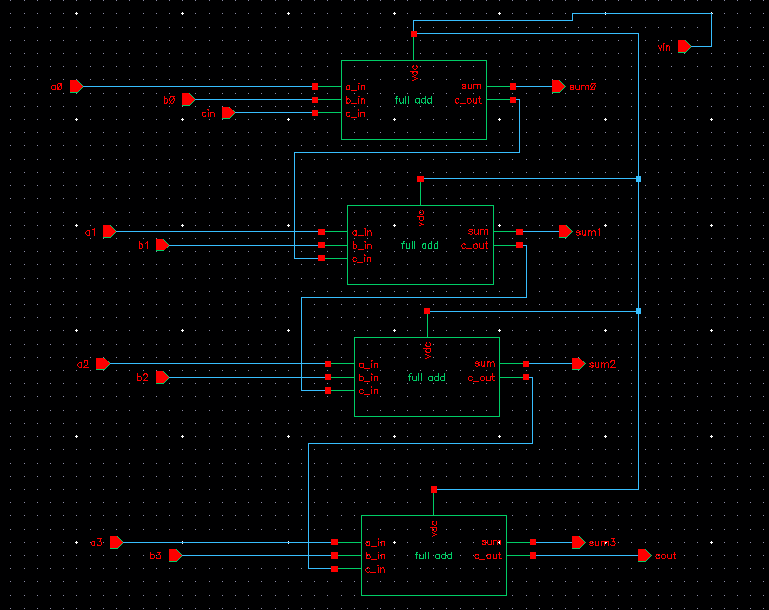


Fig 3.7: 4-bit ripple carry adder

* + 1. **4×4 Vedic multiplier**

With the use of 2-bit Vedic multipliers and adders, a 4-bit Vedic multiplier may be created. This partial product term is formed using a 2-bit Vedic multiplier, proceeded by such a partial reduction and terms of adding step using a simple adder circuit. It is feasible to utilize a straightforward 4-bit RCA circuit. Consider that A and B are the two four-bit values, respectively: A= "a3 a2 al a0" and B= "b3 b2 b1 b0". "p7 p6 p5 p4 p3 p2 pl po" is the output for the multiplication result. Say "a3 a2" & "al a0" for A and "b3 b2" & "bl b0" for B when dividing A and B into two parts. Now it is assumed that the number of bits to be multiplied by the 2x2 vedic multiplier is two bits added together and regarded as a single bit. Four bits will be the outcome of the multiplication. The first two inputs of the 2x2 bit multiplier are "al a0" and "bl a0" (Vertically). The final block is a 2x2 bit multiplier using the inputs "a3 a2" and "b3 b2" (Vertically). The two 2x2 bit multipliers with inputs "a3 a2" and "bl b0" and "al a0" and "b3 b2" are in the center (Crosswise). In the end, there are 8 bits total: "p7 p6 p5 p4 p3 p2 pl p0" and "c0," which serves as the carry bit as shown in schematic in Figure 3.8.

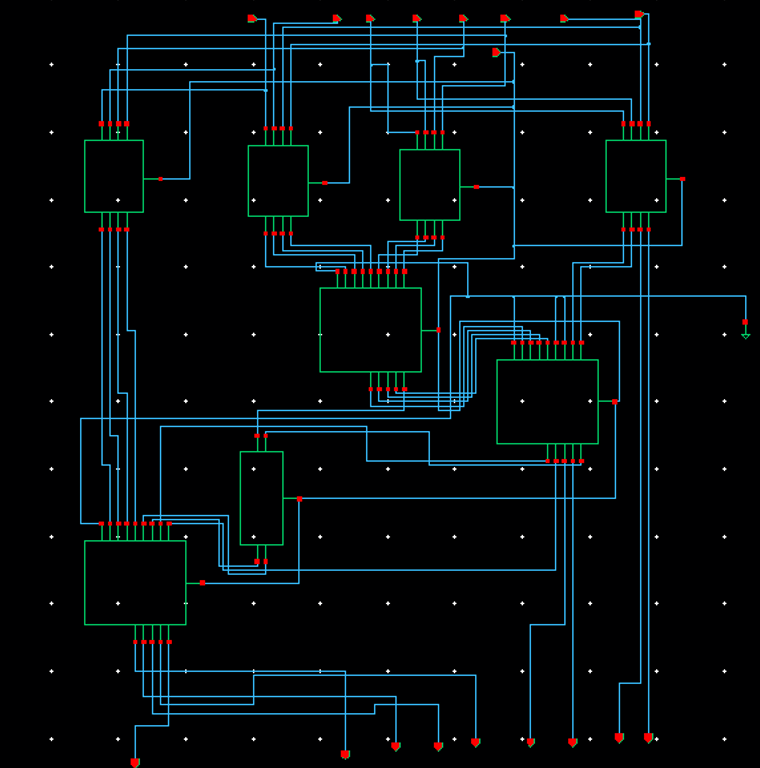


Fig 3.8: Schematic diagram of 4×4 multiplier

* + 1. **8-bit ripple carry adder**

The 8-bit ripple carry adder is made up of two 4-bit ripple carry adders. The schematic diagram is as shown below in the figure 3.9.

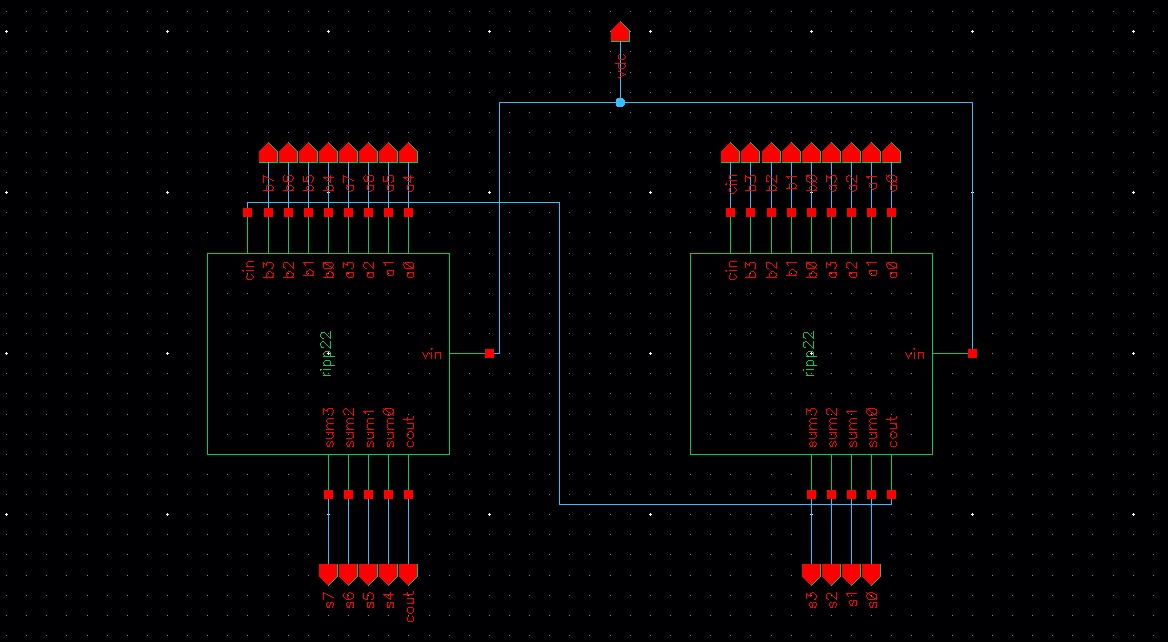


Fig 3.9: Schematic of 8-bit ripple carry adder

1. **Proposed Design**

A 4:2 Compressor is shown in figure 4.1 which is capable of adding 4 bits and one carry and produce 3 bits output. We have designed the 4:2 compressor by using full and half adder. It can be clearly seen; the critical path is smaller in comparison with an equivalent circuit to add 5 bits using full adders and half adders. We consider the propagation delay of each gate is t and we know that the propagation delay for half adder is t and 2t for full adder.

****

Fig 4.1: Block diagram of 4:2 compressor

The 5:2 compressor is capable of adding 5 bits input and two carry at a time and produce 4 bits output as shown in fig.4.3 and fig.4.4. The compressor performs a reduction operation, compressing a 5-bit binary input into a 2-bit output. The whole compressor can be used to replace the equivalent of full adder and half adder. By doing this we can reduce the critical path. Our design is focused on 5:2 compressor using full adders as shown in figure 4.4. This compressor is required to design the 7:2 compressor, which is shown in figure 4.5.



Fig 4.3: Block diagram of 5:2 compressor



Fig 4.4: 5:2 compressor based on full adder

The 7:2 compressor is capable of adding 7 bits number and two carry from previous stages at a time as shown in figure 4.5. In our purposed model, we have designed a new 7:2 compressor utilizing one 5:2 compressor and two full adders. The architecture for same is shown in figure 4.5. As we know that the 5:2 compressor increase the speed due to small critical path. Through the experiment of Xilinx Spartan-3, it was found that the new 7:2 compressor adder architecture is 1.09 time faster than a conventional approach. This result justifies the need of utilizing this compressor in our design.



Fig 4.5: 7:2 compressor based on 5:2 compressor and full adder

The partial products are generated using and gate and each of them are sent through different blocks for processing the product. The compressor architecture is designed as mentioned before in the section.

1. **Tool used** 
   1. **CADENCE VIRTUOSO**

In order to thoroughly explore, evaluate, and validate a design against by the user's intended goals, the Virtuoso Analog Design Environment product suite offers all the necessary capabilities. This enables designers to preserve design intent throughout the designing place Cadence Virtuoso System Design Platform is a comprehensive, system-based solution that has the capability to drive simulation and LVS layout of ICs and packages from a unified schematic. The leading supplier of EDA and semiconductor IP is Cadence. The transistors, common cells, and IP blocks that makeup SoCs are designed by engineers with the aid of custom/analog tools. At the most recent semiconductor processing nodes, the digital tools automate the design and verification of giga-scale, giga-hertz SoCs. Complete boards and subsystems can be designed using IC packaging and PCB tools. The design IP and verification IP portfolio offered by Cadence for memories, interface protocols, analog/mixed-signal components, and specialized processors are also expanding. Cadence provides a comprehensive set of hardware/software co-development platforms that extends up to the systems level.

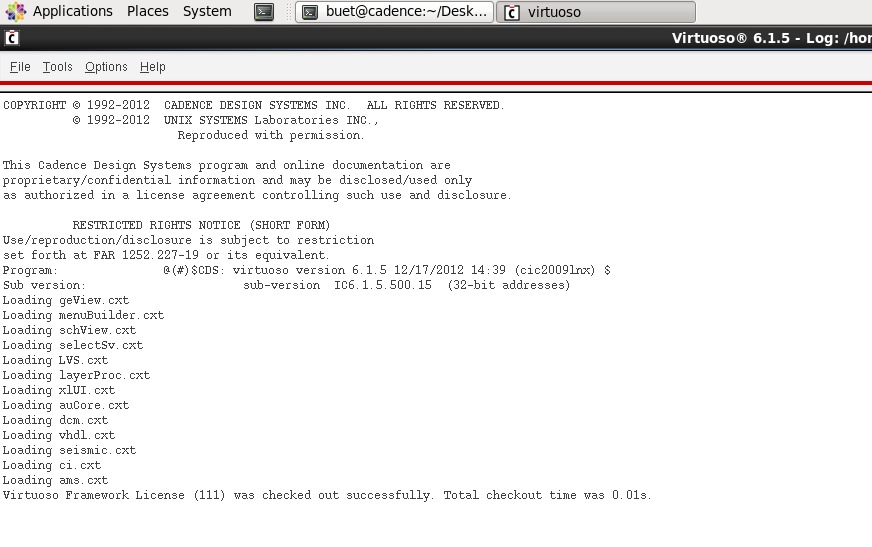


Fig 5.1: Cadence Virtuoso

* 1. **Virtuoso Analog Design Environment L.:**

The admittance electronic development and simulation framework for the Virtuoso custom design platform is called Virtuoso Analog Design Environment L. An environment for modelling and assessing fully customized, analogue, and RF IC designs is called Analog Design Environment L. It has several connectors to third-party simulators, a stream processing system, an interconnected waveform display, and a graphical user interface. Benefits include the following:

* Accelerated learning curve due to a simulator-free environment.
* Maximum productivity while using script-driven mode.
* For interactive analysis, close integration with Virtuoso Schematic Editor.
* Fast circuit exploration with simple design and test parameterization.
* Flexible window for the best display of pertinent facts.

1. **Results and Discussions**

Our main objective is to achieve a high speed, energy efficient 8-bit Vedic multiplier using compressor. We have taken the conventional method of multipliers into consideration to show the difference and betterment of our designed circuit. The conventional method we considered include the design of multipliers using adders- half adders, full adders and ripple carry adders. The procedural flow includes designing of 2×2 multiplier, 4×4 multiplier then 8×8 multiplier.

* 1. **2-bit Vedic multiplier**

The 2-bit multiplier output waveform shown in Figure 6.1 and, for inputs of "al a0" and "bl b0". Various test cases are generated for the inputs by pulse waveforms, each of which has a different pulse width and time. The outputs are represented by "p3 p2 pl p0", Where "p3" is the MSB and "p0" is the LSB.

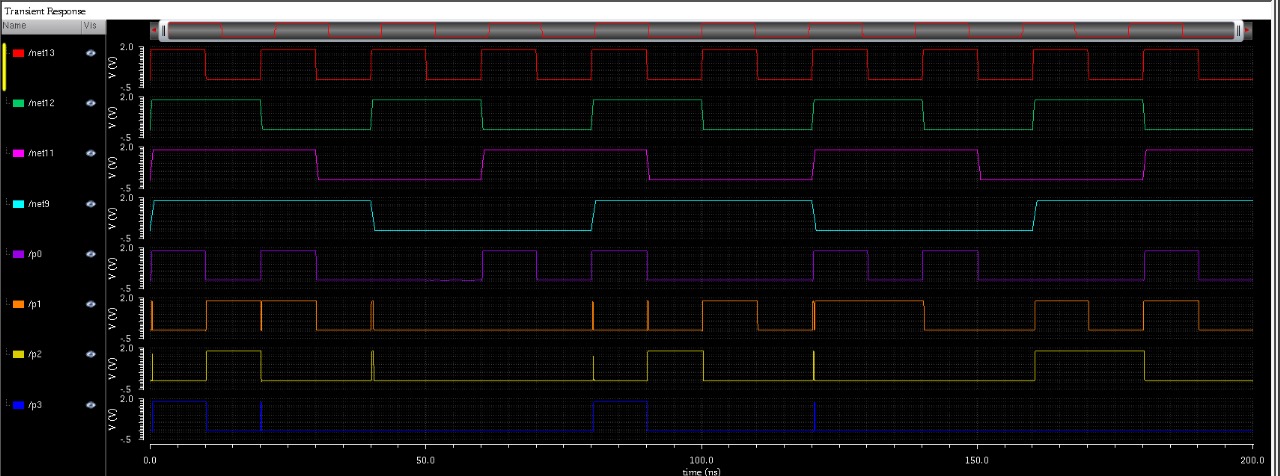


Fig 6.1: Output waveform of 2-bit multiplier

Table 6.1 Comparison between power and delay for 2-bit multiplier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Power (mW)** | | **Delay (ps)** | |
| **MOSFET** | **FinFET** | **MOSFET** | **FinFET** |
| **P3** |  |  |  |  |
| **P2** |  |  |  |  |
| **P1** |  |  |  |  |
| **P0** |  |  |  |  |

* 1. **4-bit Vedic multiplier**

The 4-bit Vedic multiplier output waveform is shown in Figure 6.2, "a3 a2 al a0" and "b3 b2 bl b0" are the two 4-bit that are multiplied and the resultant will be 8-bit "p7 p6 p5 p4 p3 p2 pl p0". The table 6.2 shows the comparison of power and delay for two different materials.

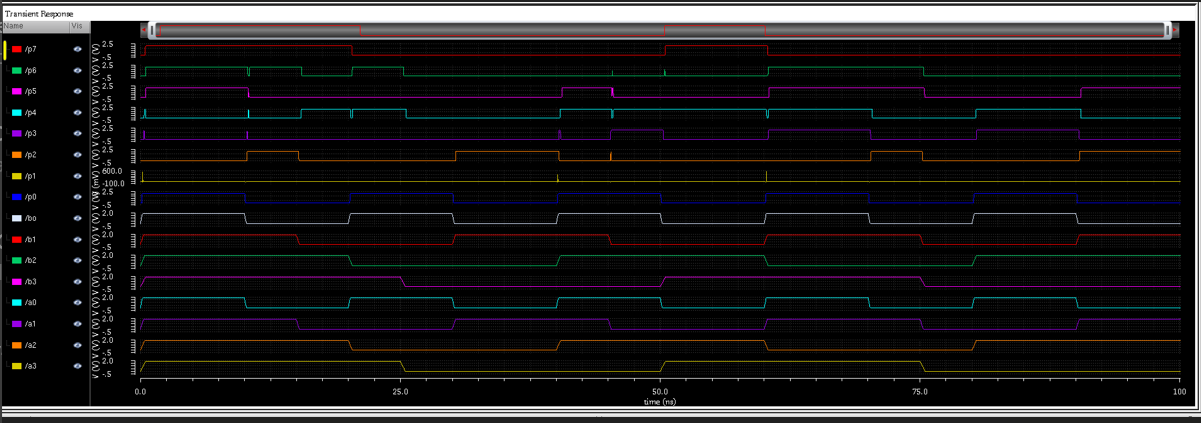


Fig 6.2 Output waveform of 4-bit Vedic multiplier

Table 6.2 Comparison between power and delay for 4-bit multiplier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Power (mW)** | | **Delay (ps)** | |
| **MOSFET** | **FinFET** | **MOSFET** | **FinFET** |
| **P7** |  |  |  |  |
| **P6** |  |  |  |  |
| **P5** |  |  |  |  |
| **P4** |  |  |  |  |
| **P3** |  |  |  |  |
| **P2** |  |  |  |  |
| **P1** |  |  |  |  |
| **P0** |  |  |  |  |

* 1. **8-bit Vedic multiplier**

The 8-bit Vedic multiplier schematic is shown in Figure 6.3, "a7 a6 a5 a4 a3 a2 al a0" and "b7 b6 b5 b4 b3 b2 bl b0" are the two 8-bit that are multiplied and the resultant will be 16-bit "p15 pl4 p13 p12 p11 p10 p9 p8 p7 p6 p5 p4 p3 p2 pl p0". The waveform of 8-bit Vedic multiplier are shown in Figure 6.4.a, 6.4.b,

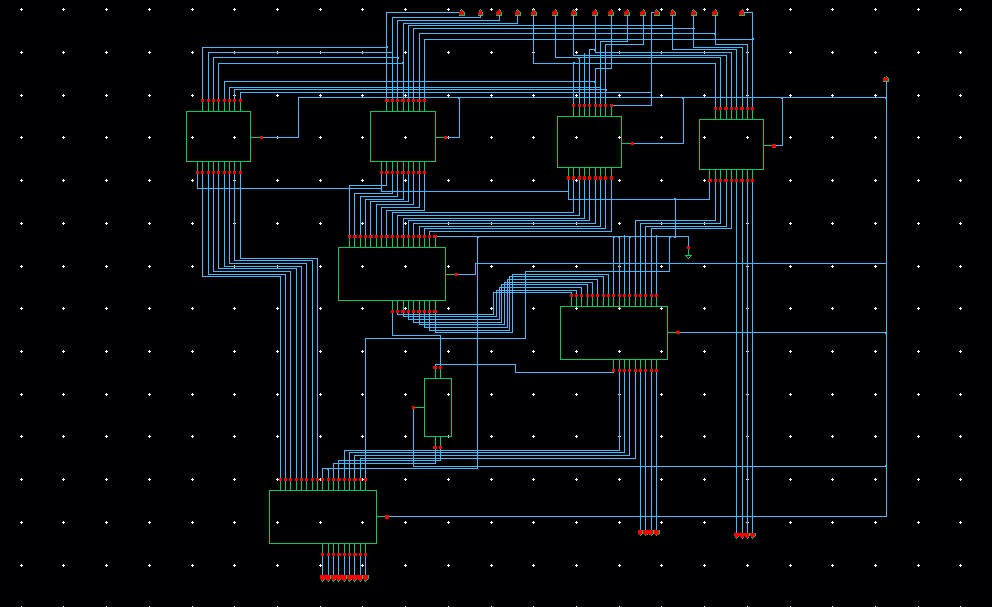


Fig 6.3: Schematic of 8-bit Vedic multiplier

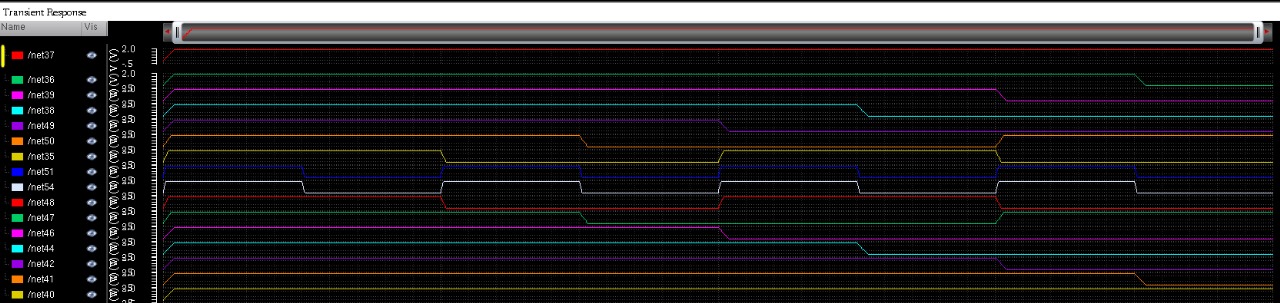


Fig 6.4.a: Output waveform showing input a (a0-a7) bits and b (b0-b7) bits

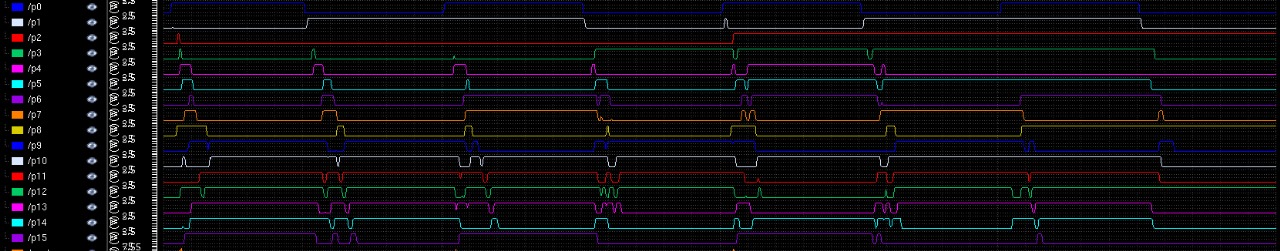


Fig 6.4.b: Output waveform showing multiplication output bits (p0-p15)

Table 6.3 Comparison between power and delay of 8-bit multiplier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Power (mW)** | | **Delay (ps)** | |
| **MOSFET** | **FinFET** | **MOSFET** | **FinFET** |
| **P15** |  |  |  |  |
| **P14** |  |  |  |  |
| **P13** |  |  |  |  |
| **P12** |  |  |  |  |
| **P11** |  |  |  |  |
| **P10** |  |  |  |  |
| **P9** |  |  |  |  |
| **P8** |  |  |  |  |
| **P7** |  |  |  |  |
| **P6** |  |  |  |  |
| **P5** |  |  |  |  |
| **P4** |  |  |  |  |
| **P3** |  |  |  |  |
| **P2** |  |  |  |  |
| **P1** |  |  |  |  |
| **P0** |  |  |  |  |

1. **Applications, Disadvantages, and Future Scope**

**Applications**

* Digital Signal Processing (DSP): In DSP applications, rapid and efficient multiplication operations are crucial for tasks such as filtering, convolution, and signal modulation. The 8-bit Vedic multiplier's parallelism and compressor architecture can significantly accelerate these operations, making it well-suited for DSP algorithms in fields like audio processing, telecommunications, and image processing.
* Image and Video Processing: Image and video processing involve extensive mathematical operations, including pixel manipulations, transformations, and filtering.
* Cryptographic Algorithms: Cryptographic algorithms often rely on modular arithmetic and multiplication operations.
* Artificial Intelligence (AI) and Machine Learning (ML): In AI and ML applications, matrix operations and vector manipulations, including dot products, are prevalent. The 8-bit Vedic multiplier, by virtue of its parallel processing capabilities and optimized critical path delays, can contribute to faster execution of neural network operations, enabling more efficient training and inference in machine learning models.
* Embedded Systems and IoT Devices: In resource-constrained environments such as embedded systems and Internet of Things (IoT) devices, where power efficiency is critical, the 8-bit Vedic multiplier's ability to reduce power consumption is advantageous.
* Communication Systems: Communication systems, including wireless communication and networking, often involve complex mathematical operations that benefit from efficient multipliers.
* Scientific Computing: Scientific simulations and computations often involve extensive matrix multiplications and numerical simulations. The parallelism inherent in the 8-bit Vedic multiplier, coupled with its reduced critical path delays, can enhance the efficiency of scientific computing tasks in areas such as simulations, computational physics, and numerical analysis.
* High-Performance Computing (HPC): In the realm of HPC, where computational speed and efficiency are paramount, the 8-bit Vedic multiplier can contribute to accelerating various arithmetic operations.

**Disadvantages**

* For complex multiplication system becomes complex.
* Need More mental calculations: Vedic problem-solving requires greater mental calculations.
* Not easy to cross check: Since Vedic mathematics primarily uses algorithms, it is challenging to double-check the solutions.

**Future Scope**

* Design the 16-bit Vedic multiplier.
* Reduce the power and delay little more.

1. **Conclusions**

In this section, a brief conclusion of the work that is going to be done / undertaken has to be presented. The write-up must end with the concluding remarks-briefly describing innovations in the approach for how you are going to implement the taken up, main achievements and also any other important feature that makes the system stands out from the rest. This is one of the most important chapters and should be carefully written. Here, you evaluate your study, state which of the initial goals was reached and which not, mention the strong and weak points of your work, etc. You may point out the issues recommended for future work also here. State these clearly, in point-wise form if necessary, with respect to the original objective.

**References**

1. Raj Kumari and Rajesh Mehra, "Power and Delay Analysis of CMOS Multipliers using Vedic Algorithm", IEEE, 978-1-4673-8587-9/16, 2019.
2. Chiranjit R Patel, Vivek Urankar, Vivek B A, V Keshav Bharadwaj, "Vedic Multiplier in 45nm Technology", IEEE, 978-1-7281-4889-2/20, 2020.
3. S. Karunakaran and B. Poonguzharselvi, "VLSI Architecture of an 8-bit Multiplier using Vedic Mathematics in 180nm Technology", IJATE, Vol. 10, Issue 3, 2017.
4. S. Prema, Ramanan S.V, R Arun Sekar, Rajan Cristin, "High Performance Reversible Vedic Multiplier Using Cadence 45nm technology", IJRAD. Volume 02, Issue 04, pp. 64-71, October 2018.
5. Malti Bansal, Jasmeet Singh, “Comparative Analysis of 4-bit CMOS Vedic Multiplier and GDI Vedic Multiplier using 18nm FinFET Technology”, Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC 2020), DOI: 10.1109/MIXDES.2015.7208523.
6. Nehru Kandasamy, Firdous Ahmad, Shashikanth Reddy, Ramesh Babu M, Nagarjuna Telagam, Somanaidu Utlapalli, "Performance evolution of 4-b bit MAC unit using hybrid GDI and transmission gate-based adder and multiplier circuits in 180 and 90 nm technology", 0141-9331, Elsevier B.V, 2018.
7. Radheshyam Gupta, Rajdeep Dhar, “Design of high performance 8 bit Vedic multiplier using Compressor”, ICAET 2020, DOI: 10.1109/ICAET.2014.7105239
8. Medhi Ghasemzadeh, Amin Akbari, “A New Ultra High Speed 7:2 Compressor with a New Structure”, Proceedings of the 22nd International Conference, DOI: 10.1109/ICOSEC49089.2020.9215317
9. M Jurczak, N Collaert, T Hoffmann, “Review of FinFET Technology”, DOI: 10.1109/SOI.2009.5318794

**Month-wise flow of events / Project Schedule**

Once, the problem is defined along with the name - title of the project of the proposed work, then a layout should be set as to how you will tackle the problem in stages, i.e., in the form of monthly progress, i.e., you have to mention the objective of each stage (monthly review objective) in the work statement column.

Mention the main points here how you have progressed your work in the beginning of the 7th semester and in the subsequent months, which is the monthly progress (normally at the end of each month).Some of the agendas / points about how you are going to plan or execute your proposed project / dissertation work should be mentioned against the respective months mentioned below in the table of flow of events. The agendas may be …………

• Searching of the project topic,

• Searching for the venue for pursuing the project work,

• Literature survey,

• Definition of the problem,

• Synopsis formulation,

• Allocation of the guide,

• Consultations with the guide,

• Going to industry for consultations,

• Getting approval from the guide / UG coordinator / HOD / Principal,

• Frequent interactions,

• Developing Some Simulation Models for the Proposed Project Work,

• Proposing a Block Diagram,

• Writing Data Flow Diagrams,

• Flow-Charts, Algorithms,

• Doing Some Coding,

• Doing Some Experiments in the Industry or in the College Lab,

• Attending Conferences,

• Attending Workshops Relating to the Project Work,

• Developing Hardware for the Proposed Work,

• Doing Some Experimental Evaluation or Validation of the Simulated Work,

• Publication of a Review Paper in a National Conference & International Conference,

• Publication of a Research Paper in National Journal & Inter-National Journal,

• Frequent Project Reviews, Stage-1, Stage-2, Stage-3, Etc…, Final Project Review,

• Comparing the Work Done With Some Standard Methods Done By Various Authors,

• Conclusion of the Project Work,

• Report Submission for Correction to the Guide,

• Thesis Submission @ the End of the Period,

• Developing Research Monograph,

• Applying for Patent,

• Final Project Work Submission.

• Attending the Viva-Voce.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Work statement | Oct | Nov | Dec | Jan | Feb | Mar | Apr | May | Jun |
| 1 | Project survey & problem identification & problem formulation, study of various technologies relating to the work, submitting the synopsis abstract |  |  |  |  |  |  |  |  |  |
| 2 | Literature survey |  |  |  |  |  |  |  |  |  |
| 3 | Data collection & developing some algorithms-1 (s/w) |  |  |  |  |  |  |  |  |  |
| 4 | **First phase-1 report submission**&  Review of project phase-1 |  |  |  |  |  |  |  |  |  |
| 5 | Development of algorithms-2  Hardware design  Software design |  |  |  |  |  |  |  |  |  |
| 6 | Development of algorithms-3  Software design |  |  |  |  |  |  |  |  |  |
| 7. | Development of algorithms-4  Checking the performance of the project done & overall demonstration of working item |  |  |  |  |  |  |  |  |  |
| 8. | Submission of the project work thesis along with working model demo (h/w – s/w) |  |  |  |  |  |  |  |  |  |
| 9. | Final viva-voce in examination |  |  |  |  |  |  |  |  |  |