Semiconductor Optoelectronics
ISSN: 1001-5868

DESIGN, IMPLEMENTATION, AND ANALYSIS OF 4 X 4-BIT VEDIC MULTIPLIER USING MGDI TECHNIQUE AT 90NM

Sneha C. Kharade
MIT-WPU, School of Electronics Engineering, Survey No, 124, Paud Rd, Kothrud, Pune,
Maharashtra 411038, E-mail: 1032200450@mitwpu.edu.in

Anuja Askhedkar
MIT-WPU, School of Electronics Engineering, Survey No, 124, Paud Rd, Kothrud, Pune,
Maharashtra 411038, E-mail: anuja.askhedkar@mitwpu.edu.in

Abstract

The major Concern for the semiconductor industry is the power dissipation and operational speed of the digital circuits. Generally, a conventional CMOS technique is used for implementing the circuits, but it is increasing the power consumption and the area of the circuit. The low-power design technique which is known as mGDI (modified GDI) can be used for implementing high-speed operational circuits with minimum delay. This technique reduces the transistor count and power consumption. The proposed implementation of 4x4 bit Vedic Multiplier simulated with 90nm gpdk and resulted with 532pSec of processing delay giving power delay product of 0.7128aJ with power dissipation of 1.34 nWatt.

Keywords: Vedic multiplier · CMOS · MGDI · 90nm gpdk · cadence · low power dissipation

1. Introduction

Over the years, VLSI technology has advanced, improving chip performance in terms of the two fundamental restrictions of latency and power. Digital circuits are the most important blocks of any system. The performance of these blocks decide the overall performance of the chips. Various Microprocessors use multiplier block. To overcome the slow and laborious nature of conventional multiplication, circuit designers must create quick multipliers. High-speed multiplication can be done with Vedic multipliers.

This work primarily focuses on the application of the method to combinational logic circuits and the findings of experimental delay measurements. Utilizing logical circuits like the AND gate, EXOR gate, full adder, and half adder, a 4*4 array multiplier can be created. The Gate Diffusion Input (GDI) and CMOS design methodologies are used to create the multiplier. The multiplier is implemented in the cadence virtuoso tool. This study compares the performance of a GDI and CMOS-based 4*4 array multiplier. The comparison of GDI and CMOS techniques will demonstrate which has the best power delay and area.

The area is always a concern when designing CMOS circuits; to reduce this, the Gate Diffusion Input (GDI) approach can be used. In this study, a fast speed 4x4 Vedic multiplier using GDI and CMOS technology is demonstrated. In comparison to a traditional CMOS multiplier, the suggested multiplier shows better performance.

2. Literature review

Power constraints, area, and delay product in VLSI de-signing are major concerns. To overcome these problems, Meti et al. [1] have designed an 8-bit Vedic multiplier with the help of mGDI technology. The use of modified Gate Diffusion Input (mGDI) 180 nm technology enabled the authors to reduce the area, delay, and power consumption by a significant amount.

A 4-bit Vedic multiplier has been designed by Patel et al. [2] using the 45nm CMOS technology. By designing the schematic and layout, a high-speed Vedic multiplier has been designed which can be used in filters and DSP processors.
Yadav et al. [3] have tried to overcome the issue of low-speed operation in conventional multipliers by de-signing a 4 X 4 Vedic multiplier. This Vedic multiplier was designed using the GDI technology to reduce the transistor count and thereby minimize the power dissipation.
Metku et al. [4] have used mGDI technique to implement Low power Null Convention Logic Multiplier which provides an insensitive approach for asynchronous designs. The reason behind choosing mGDI technique is low power and area approach using Cadence virtuoso the simulation was done with 45 nm gpdk.
Jawale et al. [5] have proposed a circuit for a hybrid Dadda Multiplier to observe the PSNR of the system. For circuit level simulation Cadence virtuoso with 45nm gpdk was used to analyse the results along with the MATLAB tool.
Desale et al. [6] proposed implementation of bit serial multiplier for multiplication in binary finite field which is based on shift and add algorithms. As 65nm has 38.21mW and 40nm has 43.52mW where the proposed 45nm has resulted in 20.35mW of power consumption.
DK et al. [7] have compared the performance of full adders based on their area and n-bit architectures of braun multiplier with implementing their schematics and layouts using cadence virtuoso with 180nm gpdk.
Patel et al. [2] gave the optimization and design of a 4-bit multiplier using Cadence Virtuoso 45nm gpdk which consumes 3.795uW of power dissipation and 250ps of delay.
Selvakumari et al. [8] proposed an 8bit Vedic Multi-plier schematic is implemented using Cadence Virtuoso with 180nm gpdk. The purpose of this design is to reduce the power consumption and transistor count to a certain level that the circuit will get a significant improvement.

Kumari et al. [9] implemented a 2-bit and a 4bit multiplier using 45nm technology using cadence virtuoso tool at backend. 2-bit multiplier consumed 55.489 pW and delay of 1.924 pS similarly the 4-bit multiplier consumed 0.0002854 watt and delay of 0.208873 µS.
Devnath et al. [9] 65nm technology PTM transistor model is used along with cadence virtuoso design suite. Models are based on 3-2 counter.
Sastry et al. [10] proposed design 8-bit Vedic multiplier is carried out the 18nm FinFET technology. The power dissipated 782.4nmW and rise of propagation delay of 1.677ns which was executed on 0.8v.
Babu et al. [11] proposed carry look ahead with mGDI technique compared with other adder

models. It has lower propagation delay. It was implemented on 90nm technology using cadence environment.

Yogendri et al. [12] With the FPGA SPARTAN 3 kit, 4*4 vedic multipliers and reversible vedic multipliers are incorporated. The VHDL/VERILOG coding process for FPGA implementation is followed by simulation to verify the logical outputs. The coding is next translated into a diagram using RTL synthesis, and lastly the design is put into practise using a fusing kit to analyse parameters including power, area, delay, voltage swing, fan-in, and fan-out.

Gupta et al. [13] In this paper using 4:2 and 7:2 compressors, a new architecture for the Vedic multiplier was introduced. This method was 2% faster and 2% smaller than the urdhava tiryagabhyam method. The timing and area of the design were tested on an FPGA from the Xilinx Spartan 3 family.

Albadry et al. [14] 4-bit multiplier using full adder cell based on full swing gate diffusion input technique. He has used cadence virtuoso simulation based on TSMC 65nm technology. The full adder he designed dissipated low power and improved the area and along with full swing output voltage.

Maitra et al. [15] proposed 2-bit multiplier using MGDI cell using Dsch38 lite software tool. He compared the proposed method with conventional method using CMOS, PTL, Transmission gate and MGDI.

Askhedkar et al. [16] have designed Modulators using the GDI method that offered significant improvement in power and delay compared to CMOS technology. Gate-level simulation with WinSpice using 0.18 µm technology confirms the utility of the presented structures.

From the above literature survey, we can conclude that Power constraints, area, and delay product in VLSI designing are major concerns. The use of modified Gate Diffusion Input (mGDI) technology enabled the authors to reduce the area, delay, and power consumption by a significant amount. By designing the schematic and layout, a high- speed optimized Vedic multiplier has been designed which can be used in filters and DSP processors. The purpose of this design is to reduce the power consumption and transistor count to a certain level so that the optimized size can be achieved.

3. Methodology

The proposed 4-bit Vedic multiplier is designed using the 90 nm CMOS technology as well as the mGDI technology. The block diagram of 4-bit Vedic Multiplier is shown in fig 3.1. The multiplier is made up of a Half Adder circuit, a full adder circuit, a Ripple Carry Adder (RCA), a multiplexer, and a 2-bit Vedic multiplier circuit.

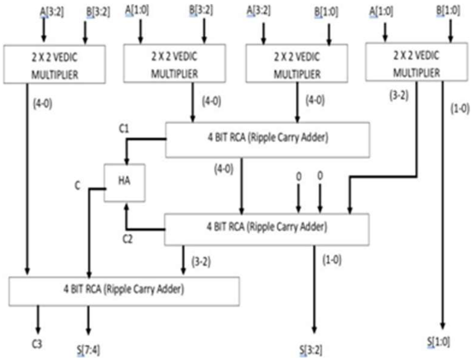


Fig.3.1 : Block Diagram of 4x4 Vedic Multiplier

3.2 Design and implementation of circuits using CMOS technology

Initially, the half adder, full adder, RCA, and 2-bit Vedic multiplier were implemented using the CMOS technology. Using all these blocks the 4-bit Vedic multiplier was designed with the same technology.

3.2.1 Two-Beit Vedic Multiplier CMOS circuit implementation

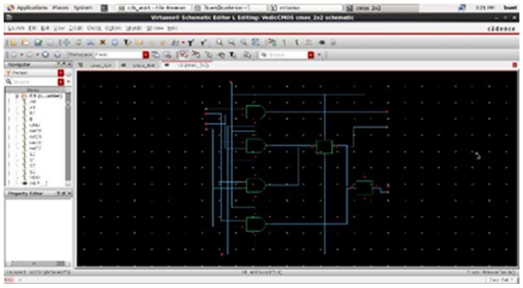

Applications Places System
cmos 2x2
Virtuoso® Schematic Editor L Editing: VedicCMOS cmos_2x2 schematic
cādence

Fig.3.2.1:Two-Bit Vedic multiplier CMOS circuit implementation

A 2-bit Vedic CMOS multiplier uses complementary metal-oxide semiconductor (CMOS) technology and the Vedic mathematical technique to multiply two 2-bit binary values.

The 2-bit Vedic CMOS multiplier's building blocks are described in the following manner: Two XOR gates: The two XOR gates generate an intermediate output by accepting the two bits from the multiplier and multiplicand as input.

Final output stage: The final output stage creates the finished product using the outputs from the two transmission gates as its input.

In particular, a full adder circuit that creates the sum and carry bits for the finished result is connected to the inputs of the two outputs.

In general, the 2-bit Vedic CMOS multiplier is a circuit that applies the Vedic mathematics technique and CMOS technology to multiple two 2-bit binary values using a combination of XOR gates, AND gates, transmission gates, and a full adder.

This kind of multiplier is effective and uses little power, making it a good choice for

applications involving digital signal processing.

3.2.2 Four-Bit vedic multiplier CMOS circuit schematic implementation

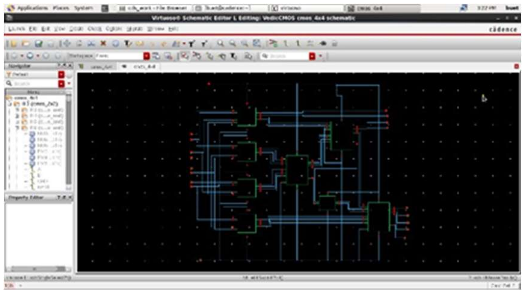

Virtuoso® Schematic Editor L Editing: VedicCMOS cmos_4x4 schematic
cādence

Fig.3.2.2 : Four-Bit vedic multiplier CMOS circuit schematic implementation

A 4-bit Vedic CMOS multiplier using 2-bit multiplier blocks is a multiplication circuit that multiplies two 4-bit binary numbers using the Vedic mathematics method and CMOS technology.

This multiplier uses two 2-bit Vedic CMOS multipliers to compute the partial products for each pair of bits in the multiplier and multiplicand.

Here is an explanation of each block in the 4- bit Vedic CMOS multiplier using 2-bit multiplier blocks:

Four 2-bit Vedic CMOS multipliers: The four 2-bit Vedic CMOS multipliers take as input two pairs of bits from the multiplier and multiplicand and produce the corresponding partial product.

Partial product generation: After the four 2-bit Vedic CMOS multipliers have computed the partial products for each pair of bits, these partial products are combined to obtain the final product.

Specifically, the four partial products are added together using a series of full adders to generate the final product.

Overall, the 4-bit Vedic CMOS multiplier using 2-bit multiplier blocks is a circuit that uses four 2-bit Vedic CMOS multipliers, full adders, and carry propagation to multiply two 4-bit binary numbers using the Vedic mathematics method and CMOS technology.

This type of multiplier is efficient, has a low power consumption, and is suitable for use in digital signal processing applications.

3.2.3. Four-bit vedic multiplier CMOS layout implementation

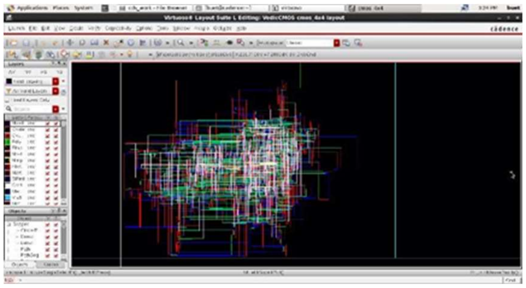


Fig.3.2.3 : Four-bit vedic multiplier CMOS layout implementation

Here are the steps to design a 4-bit Vedic CMOS multiplier layout in Cadence using a 2- bit multiplier:

Define the specifications and requirements for the multiplier, including the input and output formats, operating frequency, power consumption, and area constraints.

Design the 2-bit multiplier circuit using Vedic multiplication algorithm in CMOS technology.
Create a hierarchical design for the 4-bit multiplier, which includes two instances of the 2-bit multiplier and additional logic to combine the intermediate results.
Generate the layout for the 2-bit multiplier circuit, using the Cadence Virtuoso layout editor.
Instantiate the 2-bit multiplier layout as sub- circuits within the 4-bit multiplier layout and connect them with additional logic to combine the intermediate results.
Generate the final layout for the 4-bit multiplier circuit, using the same process as for the 2-bit multiplier.
Verify the final layout using DRC and LVS, and perform simulations to validate the functionality and performance of the complete 4-bit Vedic CMOS multiplier.

3.3 Design and implementation of circuits using mGDI technology

The mGDI technology was used to design the half adder, full adder, RCA, multiplexer, and 2-bit Vedic multi-plier. With the help of these blocks, a 4 X 4 Vedic multi-plier was designed using the mGDI technology.
A swing circuit was implemented and connected to the output pin of each block to control the voltage drop. The mGDI technology was used with the aim to reduce the circuit complexity by reducing the transistor count.

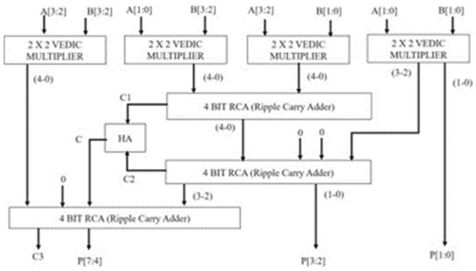


Fig.3.3 Four-Bit Vedic multiplier MGDI Block diagram

Modified Gate Diffusion Input, or MGDI, is a VLSI design technique for multiplying two numbers according to the principles of Vedic mathematics.
Bitwise operations and carry-save addition are used in the quick and effective MGDI Vedic Multiplier to multiply data. Urdhva- Tiryakbhyam, which translates to "vertically and crosswise" in Sanskrit, is the method on which it is based.
The MGDI Vedic Multiplier's fundamental design involves using a sequence of AND gates to produce partial products, then a series of XOR gates to produce the finished product. The two numbers being multiplied's bits are bitwise ANDed to create the partial products, and a series of XOR operations on the two numbers to create the final product.
Applications that call for high-speed multiplication, such digital signal processing and cryptography, benefit the most from the MGDI Vedic Multiplier. Compared to conventional multiplication methods, it has a number of benefits, such as a lower power need, a smaller footprint, and increased speed and precision.
The MGDI Vedic Multiplier, in general, is a strong and effective method that can greatly enhance the performance of digital circuits that need to execute multiplication operations.

3.3.1 Two-Bit Vedic multiplier MGDI circuit implementation

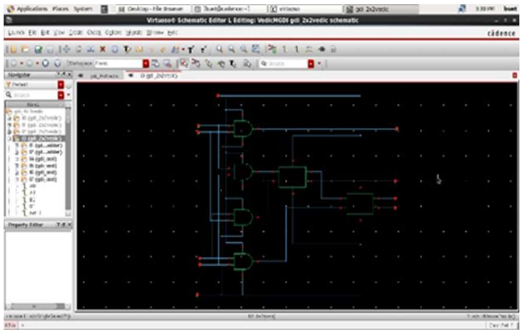

Virtuoso® Schematic Editor L Editing: VedicMGDI gdi 2x2vedic schematic
cādence

Fig.3.3.1 Two-Bit Vedic multiplier MGDI circuit implementation

The 2-bit Vedic mGDI multiplier's building blocks are explained in the following manner: Two 1-bit Half Adders: The two 1-bit Half Adders produce the total and carry bits for that bit location by taking as input the two bits of the multiplier and multiplicand. Each half adder produces a 2-bit output that reflects the sum and carry bits by performing a basic addition operation on the two input bits.

Two 1-bit Full Adders: The two 1-bit full adders accept the carry bit produced by the preceding bit position as input, along with the two bits of the multiplicand, multiplier, and multiplicand.

Carry Propagation: The carry is propagated to the next bit position by providing it as an input to the following half adder or full adder if the addition of the two half adders or full adders results in a carry.

By doing this, it is made sure that the proper carry and sum bits are generated for each bit position. Overall, the 2-bit Vedic mGDI multiplier is a straightforward and effective circuit that uses the age-old Vedic mathematics technique to quickly multiply two 2-bit binary values.

To perform Vedic MGDI multiplication with 4-bit output, we can use the following logical gate expressions for the 2x2 multiplication:

Step 1: Arrange the numbers in a grid as shown below:
A0 A1 x B0 B1

Step 2: Multiply the digits in the rightmost column of the second number (B1) with all the digits in the first number (A0, A1) using the AND gate and write the results as a row of 2 digits:
0 A0 A1 x 0 B0 B1
----------------------- 0 A0&B1 A1&B1

Step 3: Shift the second number one place to the left and repeat Step 2. Write the results in the next row below the first row:
0 A0 A1 x 0 B0 B1
----------------------- 0 A0&B1 A1&B1
0 A0&B0 A1&B0

Step 4: Add the two rows to get the result: 0 A0 A1
x 0 B0 B1
---------------------- 0 A0&B1 A1&B1
0 A0&B0 A1&B0

---------------------- P0 P1 P2 P3

The logical gate expressions for P0 to P3 are as follows:

P0 = 0
P1 = A1&B1
P2 = (A0&B1) XOR (A1&B0) P3 = A0&B0

3.3.2 Four-Bit vedic multiplier MGDI circuit schematic implementation

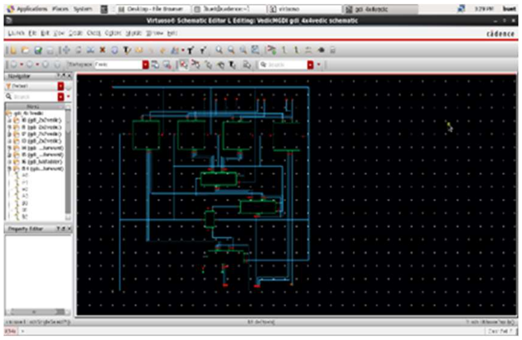


Fig. 3.3.2: Four-Bit vedic multiplier MGDI circuit schematic implementation

Using 2-bit multiplier blocks, a 4-bit Vedic mGDI multiplier can be created.

In this implementation, a 2-bit multiplier block is used to multiply each portion of the 4-bit multiplier by the matching 2-bit segment of the multiplicand. The final product is created by combining the resulting partial products. This 4-bit Vedic mGDI multiplier's blocks are each described in detail below:

Two blocks of 2-bit multipliers: The two 2- bit multiplier blocks generate the partial product for each section from the inputs of the two 2-bit sections of the multiplier and the corresponding 2-bit sections of the multiplicand.

Partial Product Generation: The partial products for each 2-bit portion of the multiplier and multiplicand are computed by the two 2- bit multiplier blocks, and then these partial products are merged to produce the final product.

Carry Generation: If the two partial products joined together to result in a carry, the carry is then added to the most important component of the final product.

Overall, this 2-bit multiplier block 4-bit Vedic mGDI multiplier is a quick and effective way to multiply 4-bit values.

To perform Vedic MGDI multiplication with 8-bit output, we need to extend the logical gate expressions for the 4x4 multiplication to include an additional row and column. The process is the same as the 4x4 multiplication, but we need to add an additional row to accommodate the extra bits in the product.

Step 1: Arrange the numbers in a grid as shown below:

0 A0 A1 A2 A3 x 0 B0 B1 B2 B3

Step 2: Multiply the digits in the rightmost column of the second number (B3) with all the digits in the first number (A0, A1, A2, A3) using the AND gate and write the results as a row of 4 digits:

0 A0 A1 A2 A3 x 0 B0 B1 B2 B3

0 A0&B3 A1&B3 A2&B3 A3&B3
Step 3: Shift the second number one place to the left and repeat Step 2. Write the results in the next row below the first row:

0 A0 A1 A2 A3 x 0 B0 B1 B2 B3

0 A0&B3 A1&B3 A2&B3 A3&B3
0 A0&B2 A1&B2 A2&B2 A3&B2
Step 4: Repeat Step 3 for the remaining digits in the second number until all digits have been multiplied and added. The result is the sum of all the rows:

0 A0 A1 A2 A3 x 0 B0 B1 B2 B3
-- 0 A0&B3 A1&B3 A2&B3 A3&B3
0 A0&B2 A1&B2 A2&B2 A3&B2
0 A0&B1 A1&B1 A2&B1 A3&B1
0 A0&B0 A1&B0 A2&B0 A3&B0
--- P0 P1 P2 P3 P4 P5 P6 P7
The logical gate expressions for P0 to P7 are as follows:
P0 = 0
P1 = A3&B3
P2 = (A2&B3) XOR (A3&B2)
P3 = (A1&B3) XOR (A2&B2) XOR (A3&B1) P4 = (A0&B3) XOR (A1&B2) XOR (A2&B1) XOR (A3&B0)
P5 = A0&B2 XOR A1&B1 XOR A2&B0 P6 = A0&B1 XOR A1&B0
P7 = A0&B0

The final product is the concatenation of the partial products P0 to P7, where P0 is the rightmost digit and P7 is the leftmost digit. Therefore, the logical gate expressions for the final product can be written as:
3.3.3. Four-bit vedic multiplier MGDI layout implementation

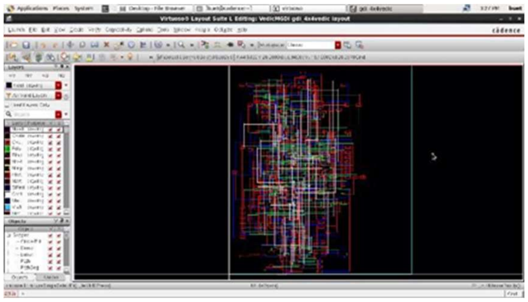

Applications Places System
Virtuoso® Layout Suite L Editing: VedicMGDI gdi 4x4vedic layout
cadence

Fig.3.3.3.: Four-bit vedic multiplier MGDI layout implementation
Here are the general steps to create a 4-bit Vedic mGDI multiplier layout using a 2-bit multiplier:

Define the specifications and requirements for the multiplier, including the input and output formats, operating frequency, power consumption, and area constraints. Design the 2-bit multiplier circuit using mGDI logic. Create a hierarchical design for the 4-bit multiplier, which includes two instances of the 2-bit multiplier and additional logic to combine the intermediate results. Generate the layout for the 2-bit multiplier circuit, using the Cadence Virtuoso layout editor. Verify the layout for the 2-bit multiplier using Cadence verification tools such as Design Rule Checking (DRC) and Layout vs. Schematic (LVS). Instantiate the 2-bit multiplier layout as sub-circuits within the 4-bit multiplier layout and connect them with additional logic to combine the intermediate results. Generate the final layout for the 4-bit multiplier circuit, using the same process as for the 2-bit multiplier. Verify the final layout using DRC and LVS and perform simulations to validate the functionality and performance of the complete 4-bit Vedic mGDI multiplier.

4. Results and Analysis:

Vedic multiplication is explained with the following two examples.

4.1 Example 1]

Multiplication of 10 and 15

I.e.. (1010 x 1111)

Then the inputs to the 4-bit Vedic multiplier will be as follows:

A3	A2	A1	A0
1	0	1	0
B3	B2	B1	B0
1	1	1	1

Table 4.1: Inputs for multiplication

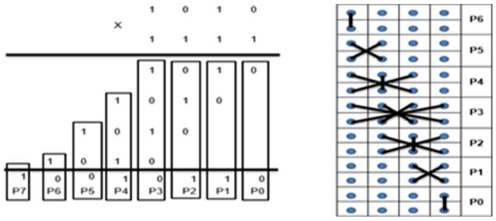


Fig.4.1.a)Multiplication calculation method

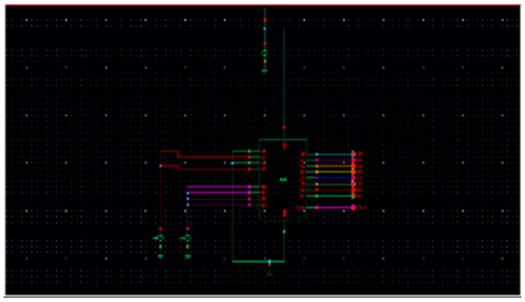
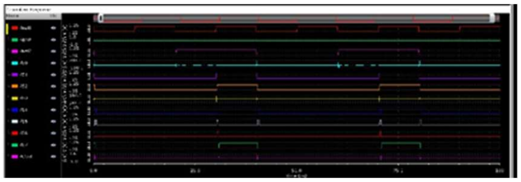


Fig.4.1.b)Multiplier test bench with above inputs

Output Waves:



Multiplication of 5 and 15
Ie. (0101 x 1111)

Then the inputs to the 4-bit Vedic multiplier will be as follows:

A3	A2	A1	A0
0	1	0	1
B3	B2	B1	B0
1	1	1	1

Table 4.2: Inputs for multiplication

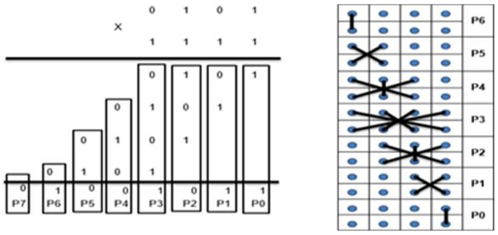


Fig.4.2.a)Multiplication calculation method

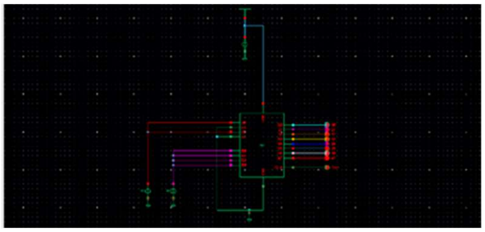


Fig.4.2.b)Multiplier test bench with above inputs

Output waves:

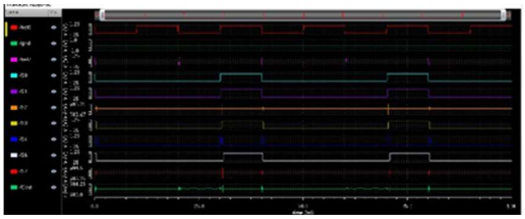


Fig.4.2.c)Multiplier output waveforms

Fig.4.1.c)Multiplier output waveforms
In above fig. output waveforms are shown, S0=0,S1=1,S2=1,S3=0,S4=1,S5=0,S6=0,S7=1 and Cout=0 ie. 150 , result can be seen for multiplication of 10 X 15 = 150.

4.2 Example 2]

In above fig. output waveforms are shown, S0=1,S1=1,S2=0,S3=1,S4=0,S5=0,S6=1,S7=0 and Cout=0 ie. 75 , result can be seen for multiplication of 5 X 15 = 75.

4.3 Example 3]

Multiplication of 15 and 15
Ie. (1111 x 1111)

Then the inputs to the 4-bit Vedic multiplier will be as follows:

A3	A2	A1	A0
1	1	1	1
B3	B2	B1	B0
1	1	1	1

Table 4.1: Inputs for multiplication

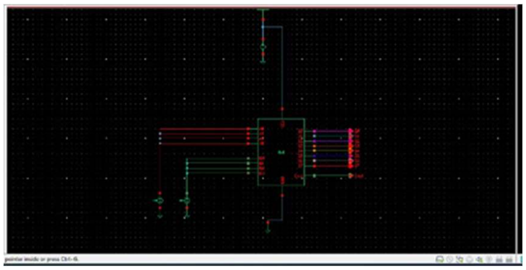
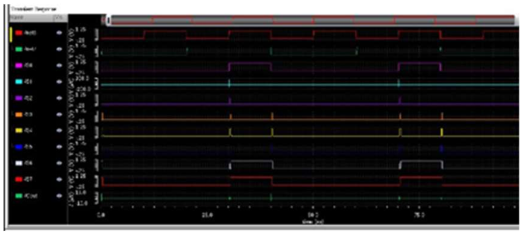


Fig.4.3.a)Multiplier test bench with above inputs

Output waves:



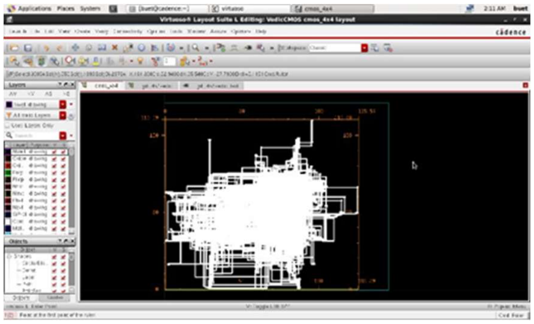


Fig.6.1. Area Calculation of CMOS

6.2. Area of Vedic mGDI 4x4 Layout

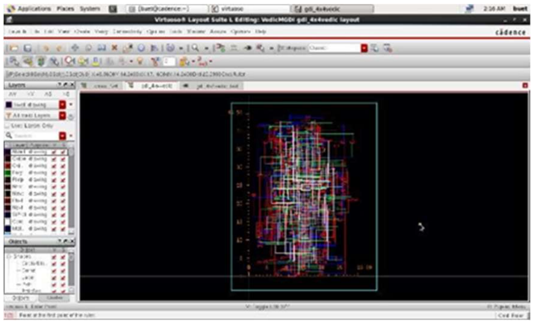


Fig.6.2. Area Calculation of MGDI

7. Applications of Vedic Multiplier

shown,
Fig.4.3.b)Multiplier output waveforms

In above fig. output waveforms are

The most important details of this multiplier are its speed, delay, and layout. It is faster than other multipliers due to the reduced area and delay caused by the two half-adders. The multiplier is unaffected by the processor's

S0=1,S1=0,S2=0,S3=1,S4=0,S5=1,S6=1,S7=1
and Cout=0 ie. 225 , result can be seen for multiplication of 15 X 15 = 225.

5. Comparison Table:

Table 5.: Comparison between CMOS and MGDI outputs for
pre-and post-layout simulation

	Pre-layout				Post-Layout		
	PD (nWatt)	Delay (pSec)	PDP (aJ)	Transistor Count	PD (nWatt)	Delay (pSec)	PDP (aJ)
MGDI	1.34	532	0.7128	226	1.54	614	0.9455
CMOS	2.40	547	1.3128	1092	2.56	667	1.7075

6. Area Calculations for Layouts

6.1. Area of Vedic CMOS 4x4 Layout

clock rate and can be expanded by expanding the input and output data bus widths. This allows designers to work around any issues to prevent disastrous device failures and layout them in microprocessors.

a) FPM application

Vedic mathematics was used to create swift floating-point multipliers, with two separate adders and a look-ahead adder. Verilog HDL is built on a Virtex-5 processor and Xilinx ISE 14.2 to the multipliers. The VMs operated faster due to partial products being created and added in comparison parallel, while the FPM, integer multiplier, and reconfigurable multiplier were also compared.

b) Utilization in image processing

Image processing techniques involve spatial filtering techniques such as image

leveling, flattening, smoothing, refining, and sharpening. The output is then displayed as an image on a Video Graphics Array screen. Image processing is used in a variety of fields, including filtering and the detection of skin cancer, with the most recent and widely used constant multiplications being SCM and MCM. The Vedic Multiplier and Ripple Carry Adder speed up filter processing.

c) Compression of Images: Implementation of the DCT algorithm in Discrete
d) 2-D DWT, with varying levels of discrete wavelet transformation
e) used in Novel Vedic Asynchronous DSP Core with VM and Divider development
f) High-Speed VM and CSLA in combination in a parallel FIR architecture
g) Used in the setting up of 64-bit IIR filters
k) Application in Hierarchy multiplier
h) Vedic methodology implementation on VLSI for cubic computation
i) signal processing in biomedicine (FinFET & CNTFET technology)

8. Conclusion

The comparison table above shows the results obtained after implementing Vedic Multiplier in Cadence tool using traditional CMOS and the current mGDI technique. The pre and post-layout simulations were carried out using both techniques in 90 nm technology. The Vedic Multiplier implemented using the mGDI technique shows improvement in terms of area (not shown in the table above), power dissipation, delay, and power delay product. There is an improvement of 40% in power dissipation, almost 8% improvement is obtained in delay and there is an improvement of 45% in the power-delay product. This Vedic Multiplier developed using the mGDI technique makes it most suitable for these applications.

It can be easily laid out with a silicon chip because of its uniform structure. Convolution, correlation, and FFT are a few applications. Applications are put into place to expand the use of DSP in VLSI. Texas uses proprietary DSP processors. Applications for discrete-time signals should be prioritized and utilized. The Vedic multiplier is built on MAC or multiplies and accumulate. They can be used in DSP in a variety of ways.

References

[1] S.S. Meti, C. Bharath, Y.P. Kumar, B. Kariyappa, in 2017 International Conference on Advances in Comput- ing, Communications and Informatics (ICACCI) (IEEE, 2017), pp. 1923– 1927.

[2] C.R. Patel, V. Urankar, B. Vivek, V.K. Bharadwaj, in 2020 Fourth International Conference on Comput- ing Methodologiesand Communication (ICCMC) (IEEE, 2020), pp. 21–26.

[3] R. Yadav, M. Kumar, in 2020 International Conference on Electrical and Electronics Engineering (ICE3) (IEEE, 2020), pp. 527–529.

[4] P. Metku, K.K. Kim, Y.B. Kim, M. Choi, in 2018 Inter-national SoC Design Conference (ISOCC) (IEEE, 2018),pp.233–234.

[5] J.A. Arjun, S. Majumdar, in 2019 IEEE 16th India Coun-cil International Conference (INDICON) (IEEE, 2019),pp.1–4.

[6] Y.G. Desale, V. Ingale, in 2019 IEEE 5th International Conference for Convergence in Technology (I2CT) (IEEE, 2019), pp. 1–4.

[7] D.K. DK, R. Shilpa, B. Kavyashree, in 2017 International Conference on Current Trends in Computer, Electri- cal, Electronics and Communication (CTCEEC) (IEEE, 2017), pp. 499–504.

[8] C. Selvakumari, M. Jeyaprakash, A. Kavitha, in 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom) (IEEE, 2016), pp. 1514–1520.

[9] B. C. Devnath, S. N. Biswas and M. R. Datta, "4-bit Wallace and Dadda Multiplier design using novel hybrid 3-2 Counter," 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), 2020, pp. 189-194.

[10] N. H. Sastry, J. B. S. Bharadwaj and G. S. Jeevith, "Design and Implementation of 8-Bit Vedic Multiplier in 18nm FinFET Technology," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2021, pp. 251-256.

[11] Babu, P.A., Nagaraju, V.S., Vallabhuni, R.R. (2022). 8-Bit Carry Look Ahead Adder Using MGDI Technique. In: Nayak, P., Pal, S., Peng, SL. (eds) IoT and Analytics for Sensor Networks. Lecture Notes in Networks and Systems, vol 244. Springer, Singapore.

[12] Yogendri and A. K. Gupta, "Design of high performance 8-bit Vedic multiplier," 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), 2016, pp. 1-6.

[13] R. Gupta, R. Dhar, K. L. Baishnab and J. Mehedi, "Design of high performance 8 bit Vedic Multiplier using compressor," 2014 International Conference on Advances in Engineering and Technology (ICAET), 2014, pp. 1-5.

[14] O. A. Albadry, M. A. Mohamed El-Bendary, F. Z. Amer and S. M. Singy, "Design Of Area Efficient And Low Power 4-Bit Multiplier Based On Full- swing GDI technique," 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019, pp. 328-333.

[15] Maitra, Subhashis. "Design and Simulation of Two–bit Multiplier Circuit using MGDI." (2019).

[16] Anuja Askhedkar, Girdharilal Agrawal,"Low Power, Low Area Digital Modulators using Gate Diffusion Input Technique",Journal of King Saud University - Engineering Sciences,Volume 31, Issue 3,2019, ISSN 1018-3639, Pages 245-252.