INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 3)
Available online at: www.ijariit.com

Design of low power delay efficient Vedic multiplier using reversible gates

B Ramya
bramyabrbg9741@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Navya Shree G
navyagarapati231@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

D Anvesh Kumar
anvesh1175@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Bapuram Harshavardhan Reddy
harshavardhanb333@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

Hari Krishna Moorthy
locushari@gmail.com
School of Engineering and Technology Jain University (SET JU), Bengaluru, Karnataka

ABSTRACT

In early days of computers, multiplication was implemented generally with a sequence of addition, subtraction and shift operations. There exist many algorithms proposed in the literature to perform multiplication, each offering different advantages and having trade-off in terms of delay, circuit complexity, area occupied on-chip and power consumption. Latency is the major issue of computing a function. Simply it's a measure of how long the inputs to a device are stable is the final result available on outputs. Throughput is the measure of how many multiplications can be performed in a given period of time. The multiplier is not only a high delay block but also a major source of power dissipation. Normal multiplication process involves generation of partial products, the addition of partial products and finally, total product is obtained. So the performance of the multiplier depends on the number of partial products and the speed of the adder. The reversible computation is one field which assures zero power dissipation. Thus during the design of any reversible circuit, the delay is the only parameter that has to be taken care of. Hence reversible Urdhva Tiryakbhayam [UT] Multiplier had been proposed for reversible calculations. Vedic multiplier based on the Urdhva Tiryakbhayam algorithms provide the best results in terms of delay, area, and power.

Keywords: *Reversible logic gate, Urdhva Tiryakbhayam, Optimised design.*

1. INTRODUCTION

Vedic mathematics is a standout amongst the oldest techniques utilized by the Aryans to perform numerical calculations. These calculations come down a substantial number of tasks to basic arithmetic calculations. The efforts put by Jagadguru Swami Sri Bharati Krishna Tirtha Maharaja to introduce Vedic Mathematics to the commoners as well as streamline Vedic Algorithms into 16 formulas or Sutras needs to be acknowledged and appreciated. The Urdhva Tiryakbhayam method is one such calculation which is outstanding for its productivity in lessening the counts included. With the advancement in the VLSI technology, there is an ever-increasing urge for portable and embedded Digital Signal Processing (DSP) systems. Multiplication is the most basic and frequently used operations in a CPU. Multiplication is an operation of scaling one number by another.

There is always a trade-off between the power dissipated and speed of operation. Normal Multiplication process involves generation of partial products, the addition of partial products and finally, total product is obtained. So the performance of the multiplier depends on the number of partial products and the speed of the adder. The reversible computation is one field that assures zero power dissipation. Thus during the design of any reversible circuit, the delay is the only parameter that has to be taken care of. Hence

reversible Urdhva Tiryakbhayam [UT] Multiplier had been proposed for reversible calculations. Vedic multiplier based on the Urdhva Tiryakbhayam algorithms provide the best results in terms of delay, area, and power.

2. LITERATURE SURVEY

Conventional combinational logic circuits are known to dissipate heat for every bit of information that is lost. This is also evident from the second law of thermodynamics which states that any irreversible process leads to loss of energy. Landauer demonstrated that any entryway that is irreversible, essentially disseminates vitality, and each irreversible piece produces k*T ln2 joules of heat where k is Boltzmann's consistent (1.38 x 10-23 joules/Kelvin) and T is the temperature in Kelvin. Bringing down the edge voltage and administration of the power supply are generally connected practices to diminish the vitality utilization in any intelligent task.k*T ln2 joules of heat where k is Boltzmann's constant (1.38 x 10-23 joules/Kelvin) and T is the temperature in Kelvin. However, these technologies of lowering the energy consumption will hit a barrier of kT. In order to reduce this, techniques such as reducing the temperature of computer and constructing a thermodynamically reversible computer can be used.

Frank analyzed that the second option was a better choice. When the temperature of the system reduces to absolute zero, the energy reduces two orders of magnitude but using reversible computing there can be further more reduction that matches with the theoretical value. The basic feature of reversible computation is that electric charge on storage cell consisting of transistors is not allowed to flow away during transistor switching. This can be reused through reversible computing and hence decrease energy dissipation. Bennett in 1973 showed that an irreversible computer can always be made reversible. Reversible logic circuits usually take care of heating because in a reversible logic every input vector can be uniquely recovered from its output vectors and hence no information is lost. A Reversible Logic gate is an n-input n-output logic function in which there is a one-to-one correspondence between the inputs and the outputs. This not only helps to determine the outputs from the inputs but also the inputs can be uniquely recovered from the outputs. Because of this bijective mapping, the output vectors are merely permutations of the input vectors.

The most unmistakable utilization of reversible rationale lies in quantum PCs. A quantum PC can be seen as a quantum organize made out of quantum rationale entryways or circuits; each door plays out a basic unitary task on one, two or more two-state quantum frameworks called qubits. Each qubit speaks to a rudimentary unit of data relating to the established piece esteems 0 and 1. Reversible Gates are circuits in which a number of yields are equivalent to the quantity of information sources and there is a balanced correspondence between the vector of data sources and yields. A reversible circuit ought to be composed utilizing least number of reversible rationale entryways. From the perspective of planning a reversible circuit, there are numerous parameters for assurance of intricacy and execution of circuits.

A circuit is said to be reversible if the info vector can be particularly recouped from the yield vector and there is a balanced correspondence between its information and yield assignments, i.e. not just the yields can be exceptionally decided from the data sources, yet additionally, the data sources can be recouped from the yields . Energy dissemination can be decreased or even dispensed with if calculation progresses toward becoming Information-lossless. A reversible rationale entryway is an n-input n-yield rationale gadget with balanced mapping. This decides the yields from the sources of info and furthermore, the information sources can be interestingly recouped from the yields. However, fan-out in reversible circuits is accomplished by utilizing extra doors. A reversible circuit ought to be planned to utilize less number of reversible rationale entryways. From the purpose of the reversible circuit configuration, there are numerous parameters for assurance of multifaceted nature and execution of circuits.

2.1 Reversible Logic Gates

Feynman Gate: It is a 2x2 gate. If the first input i.e. A is given as 1 the second output will be the complement of the second input i.e. B. so, this gate is also known as Controlled Not Gate. It can also be used to copy inputs. Quantum cost of this gate is one.

Peres Gate: It is a 3x3 gate. It can be used as a half adder with third input i.e. c as 0. It also serves the purpose of fan out. Quantum cost of this gate is four.

Haghparast and Navi gate (HNG): It is a 4x4 gate.This gate act as one-bit full adder. Quantum cost of this gate is six.
BVPPG gate: In this 5x5 reversible gate is proposed. It is basically for multiplication and can generate two partial products at a time. Quantum cost of this gate is ten.

New Fault-Tolerant gate (NFT) – is also a 3x3 gate. Quantum cost of this gate is five.

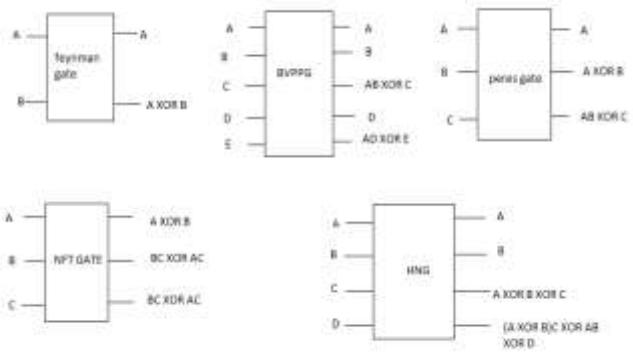

Fig -1 Reversible logic gates

3. METHODOLOGY

Consider two numbers 5 and 6.Product of 5&6 is 30.
In binary 5 represented as 101 and 6 represented as 110.
Using Urdhva Tiryakbhayam algorithm multiplication is performed as shown in below steps:

Step 1:
Take right hand digits and multiply together which gives LSB digits of the answer.

```
1 0 1
    |          Product:1x0=00
1 1 0
  0
  0 ——  Carry
```

Step 2:

Multiply LSB of the top number with second bit of bottom number and LSB of bottom number

With second bit of top number.Once we get those values, add them together.

```
1 0 1          Product:(1x1)+(0x0)=01
  X
1 1 0
  1 0
0 0 ——  Carry
```

Step 3:

Multiply LSB digit of bottom number with MSB digit of top one. LSB digit of top number

with the MSB digit of bottom and then multiply the second bit of both and

then add them all together.

```
1 0 1
  X          Product:(1x1)+(1x0)+(0x1)=01
1 1 0
1 1 0
0 0 0 ——  Carry
```

Step 4:

This step is similar to second step ,just move one place to the left.Multiply the second digit of one number by the MSB of the other number.


```
1 0 1
  X
1 1 0
-------
1 1 1 0
0 0 0 0 ———— Carry
```

Product :(1x1)+(0x1)=01

Step 5:

Multiply LSB of both the numbers together to get the final product.

```
    1 0 1
    |
    1 1 0
  ---------
  1 1 1 1 0
  0 0 0 0 0 ———— Carry
-------------
  0 1 1 1 1 0   =30
```

Product :(1x1)=01

1.5.2. Algorithm for 4X4 multiplier:

Consider two numbers 9 and 15.Product of 9&15 is 135.

In binary 9 represented as 1001 and 15 represented as 1111.

Using Urdhva Tiryakbhayam algorithm multiplication is performed as shown in below steps:

Step 1:

```
1 0 0 1
      |
1 1 1 1
-------
      1
    0 ———Carry
```

Step 2:

```
  1 0 0 1
      X
1 1 1 1
---------
    1 1
  0 0
```

Step 3:

```
  1 0 0 1
    X
1 1 1 1
---------
    1 1 1
  0 0 0
```


Step 4:

```
    1 0 0 1
      \ /
    1 1 1 1
-------------
    0 1  1 1
  1 0 0 0
```

Step 5:

```
      1 0 0 1
        \ /
      1 1 1 1
-----------------
    1 0 1  1 1
0   1 0 0  0
```

Step 6:

```
      1 0 0 1
        \ /
      1 1 1 1
-----------------
    1 0 1  1 1
0   1 0 0  0
```

Step 7:

```
        1 0 0 1
          |
        1 1 1 1
-----------------
      1 1 1 0 1 1 1
    0 0 0 1 0 0 0
-----------------------
  1 0 0 0 0 1 1 1   ——— Result =135
```

4. IMPLEMENTATION

By using logic gates and, or, xor we design buffer, reversible gates such as Feynman, Peres, BVPPG gates. Schematics of reversible gates are shown below.

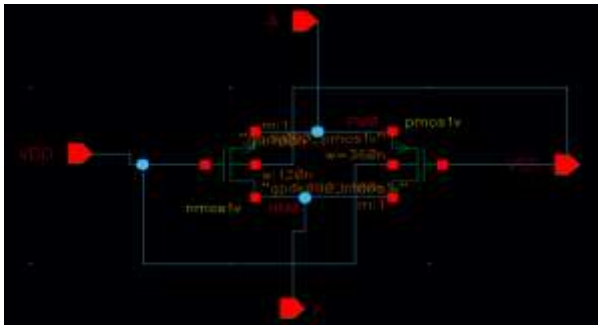

Fig -2 Schematic of Buffer Gate

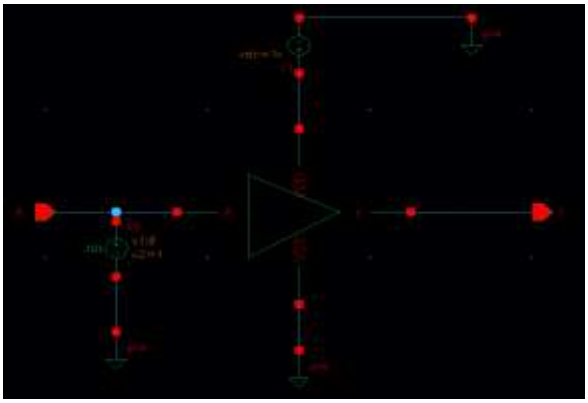

Fig -3 Symbol of Buffer Gate

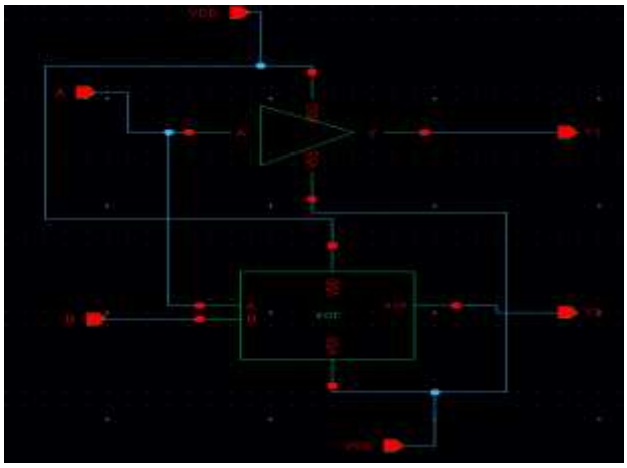

Fig -4 Schematic of Feynman Gate

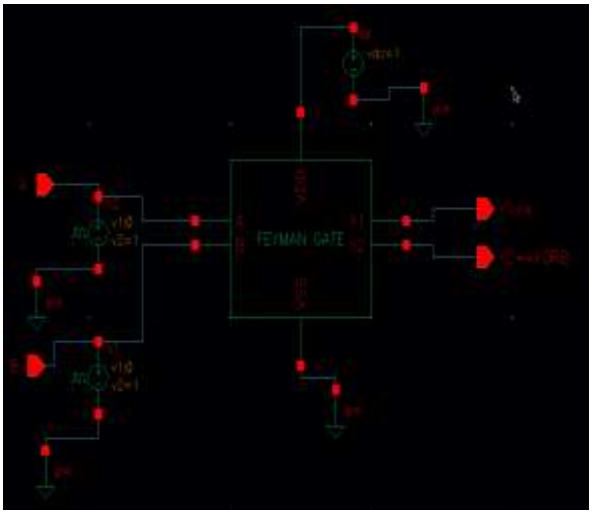

Fig -5 Symbol of Feynman Gate

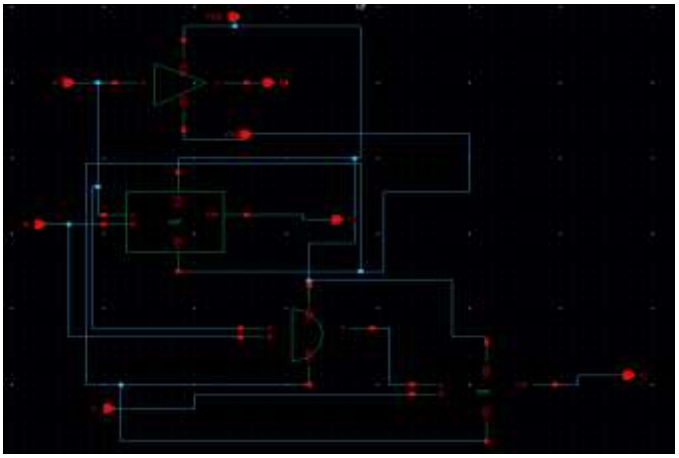

Fig- 6 Schematic of Peres Gate

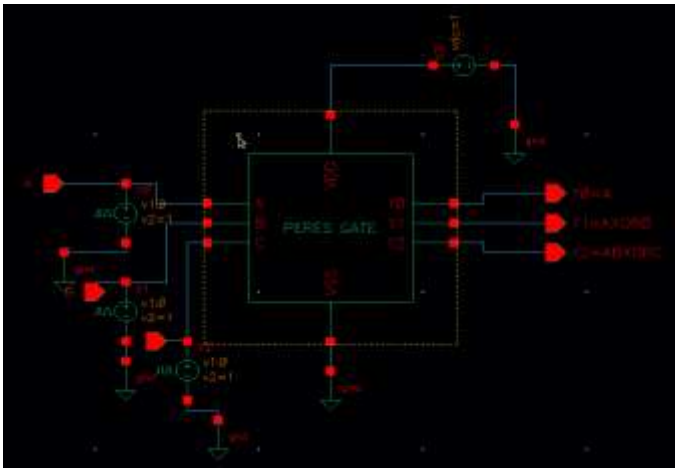

Fig -7 Symbol of Peres Gate

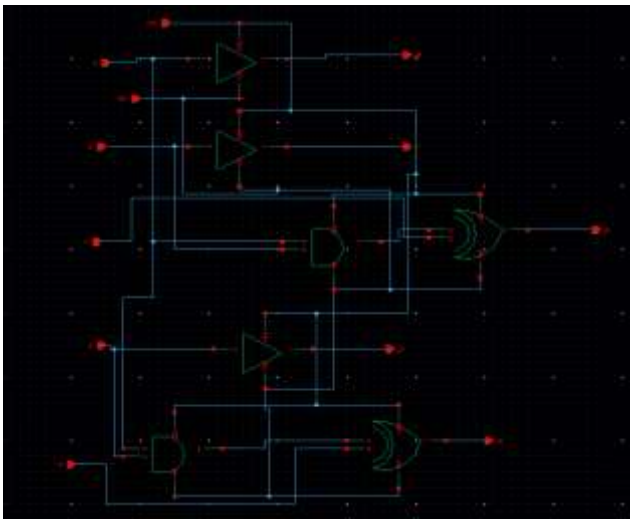

Fig-8 Schematic of BVPPG Gate

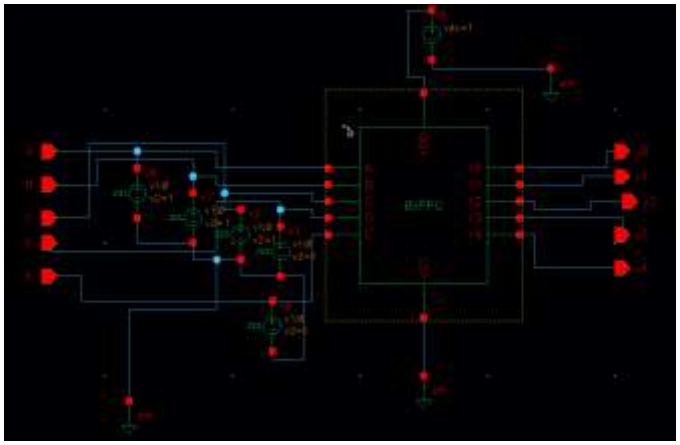

Fig -9 Symbol of BVPPG Gate

4.1 Implementation of normal binary Vedic multiplier, optimized and nonoptimized Vedic multipliers.

4.1.1: 2x2 Binary Vedic multiplier

The 2x2 Vedic multiplier is implemented using 4 equations mentioned below and the logical diagram is shown in Fig 9

q0 = a0.b0 ..... (1)
q1= (a1.b0) xor (a0.b1) ..... (2)
q2= (a0.a1.b0.b1) xor (a1.b1) ..... (3)
q3=a0.a1.b0.b1..... (4)

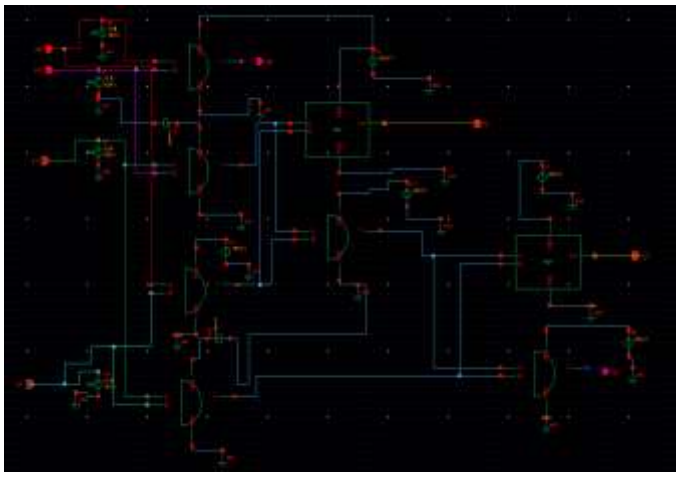

Fig -10 Schematics of Binary Vedic multiplier

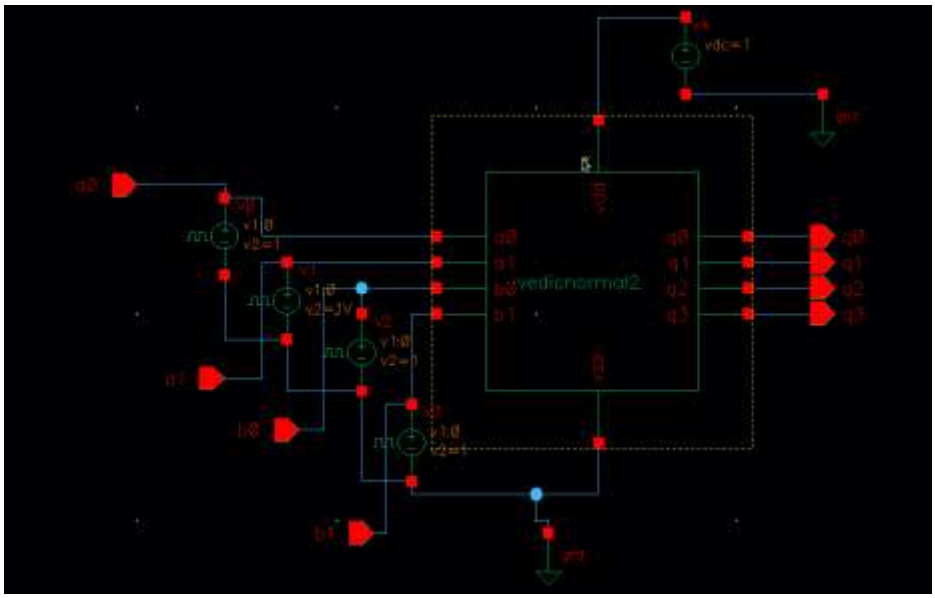

Fig-11 Symbol of Binary Vedic multiplier

4.1.2: Nonoptimized 2x2 Vedic multiplier

The reversible implementation of the circuit uses five Peres gates and one Feynman gate as shown in fig 10. This design has a total quantum cost of 21, a number of garbage outputs as 11 and number of constant inputs 4. General execution of the UT multiplier is scaled up by advancing every individual unit as far as quantum cost, garbage outputs etc.

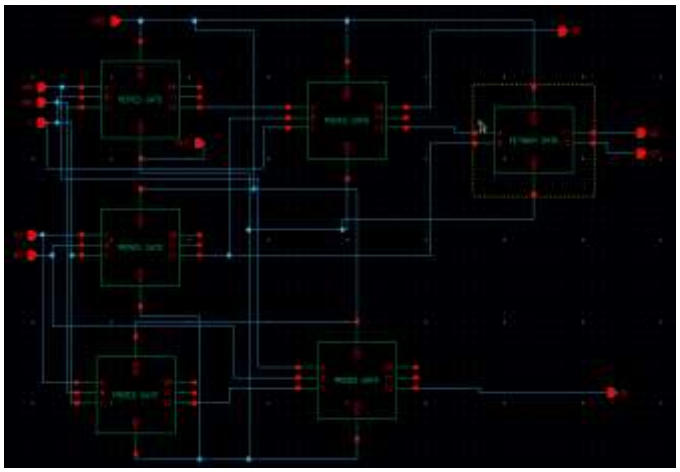

Fig -12 Schematic of Nonoptimized 2x2 Vedic multiplier

4.1.3: Optimized 2x2 Vedic Multiplier

Reversible implementation is done using a BVPPG gate, three Peres gates and a Feynman gate as shown in fig 11. BVPPG gate generates two partial products among which, one is Q0. Q1 is obtained from one of the Peres gates and Q2, Q3 are the outputs from Feynman gate. This design needs five reversible logic gates, five constant inputs and generates five garbage outputs.

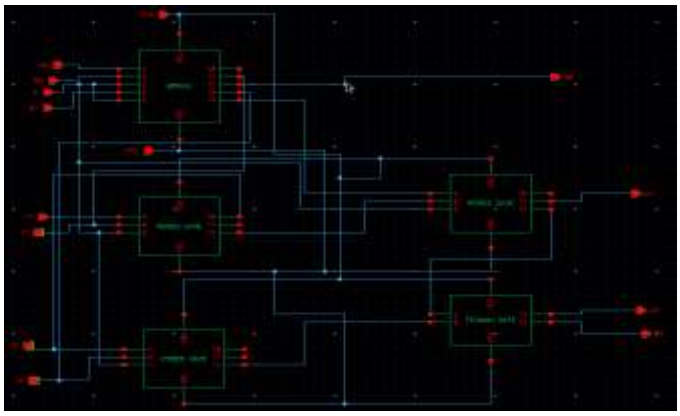

Fig -13 Schematic of Optimized 2x2 Vedic multiplier

4.1.4: Nonoptimized 4x4 Vedic multiplier

Block diagram of 4x4 is shown in Fig 12. In this block, four 2x2 multipliers are arranged systematically. Each multiplier accepts four input bits; two bits from multiplicand and other two bits from the multiplier. Addition of partial products is done using two four bit ripple carry adder and 5bit RCA

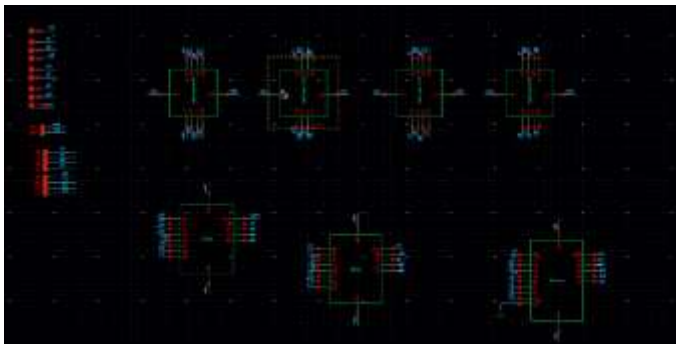

Fig -14 Schematic of nonoptimized 4x4 Vedic multiplier

4.1.5: Optimized 4x4 Vedic Multiplier

Block diagram of 4x4 is shown in Fig 13. In this block, four 2x2 multipliers are arranged systematically. Each multiplier accepts four input bits; two bits from multiplicand and other two bits from the multiplier. Addition of partial products is done using two

four bit ripple carry adder, a two-bit ripple carry adder and a half adder. We get final result by concatenating last 2 bits of the first multiplier,4 sums bits of second 4bit rca, and the sum.

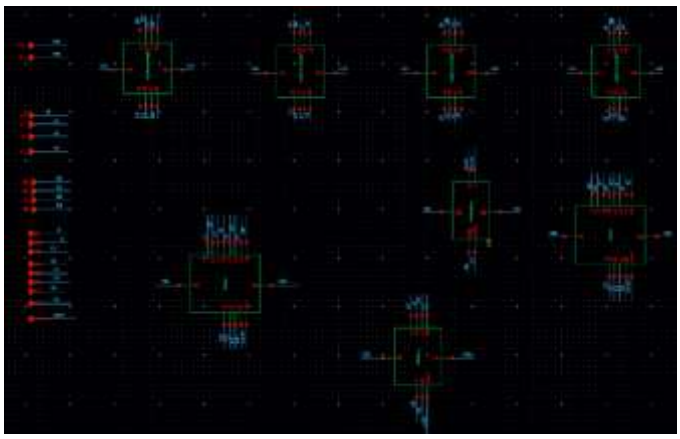

Fig -15 Schematic of Optimized 4x4 Vedic multiplier

4.1.6: 8x8 Vedic Multiplier

The Reversible 8x8 UT Multiplier design starts from the 2X2 multiplier. The block diagram of the 8x8 Vedic Multiplier is shown in the fig.14. It consists of four 2X2 multipliers each of which obtains four bits as inputs; two bits from the multiplicand and two bits from the multiplier. The lower two bits of the yield of the initial 2X2 multiplier are captured as the most reduced two bits of the last result of the multiplication. Two zeros are concatenated with the upper two bits and given as input to the four-bit ripple carry adder. The other 4 bits for RCA are obtained from the 2x2 multiplier. Likewise, the outputs of the third and the terminal 2X2 multipliers are given as inputs to the second four-bit ripple carry adder. The outputs of this 4 bit ripple carry adders are 5 bits each which should be summed up. This is done by a five-bit ripple carry add which generates a six-bit output. These six bits from the upper bits of the final result.

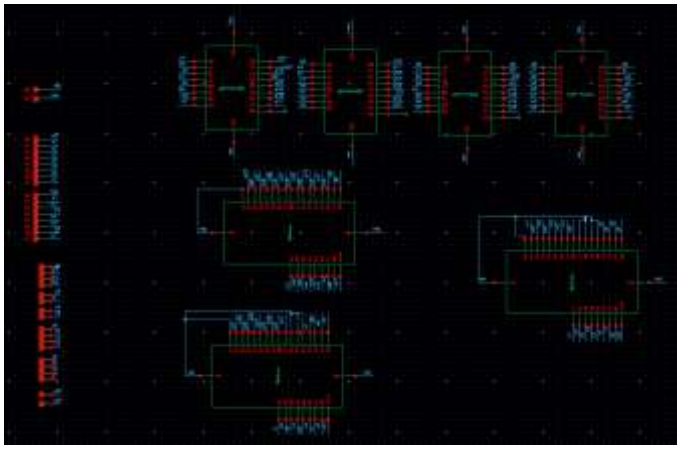

Fig -16 Schematic of 8x8 Vedic Multiplier

5. RESULTS AND DISCUSSION

The circuits are designed in cadence virtuoso environment using 45nm and 90nm GPDK tool kit with a supply voltage of 1V. The output waveforms and results of 2x2, 4x4 and 8x8 optimized and nonoptimized Vedic multipliers.

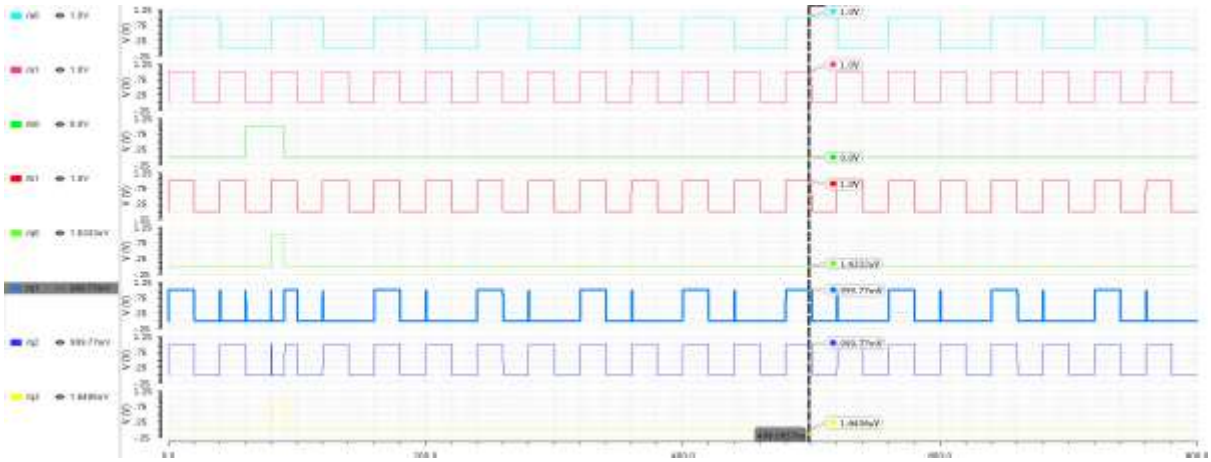

Fig -17 Output waveform of 2x2 Binary Vedic multiplier

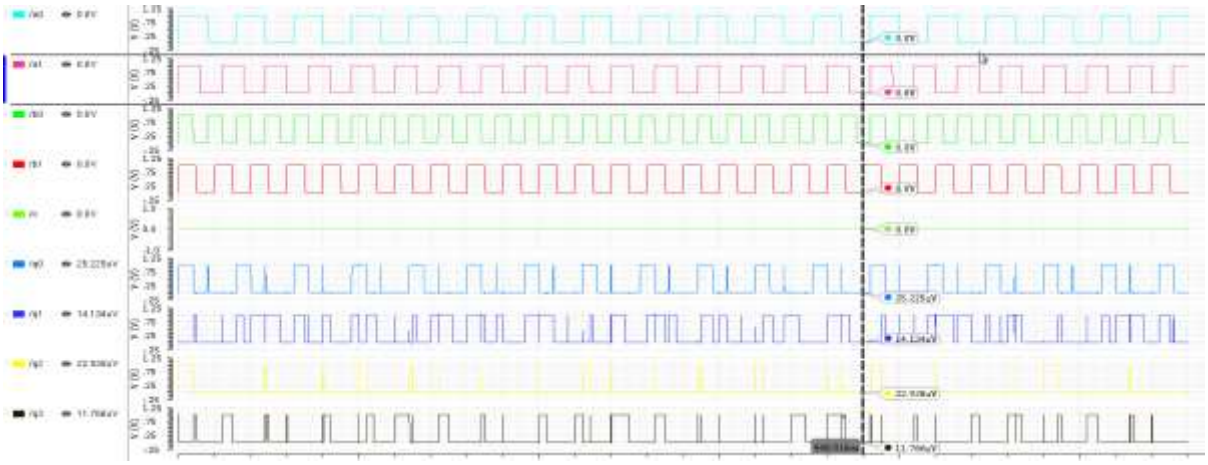

Fig -18 Output waveform of Nonoptimized 2x2 Vedic multiplier

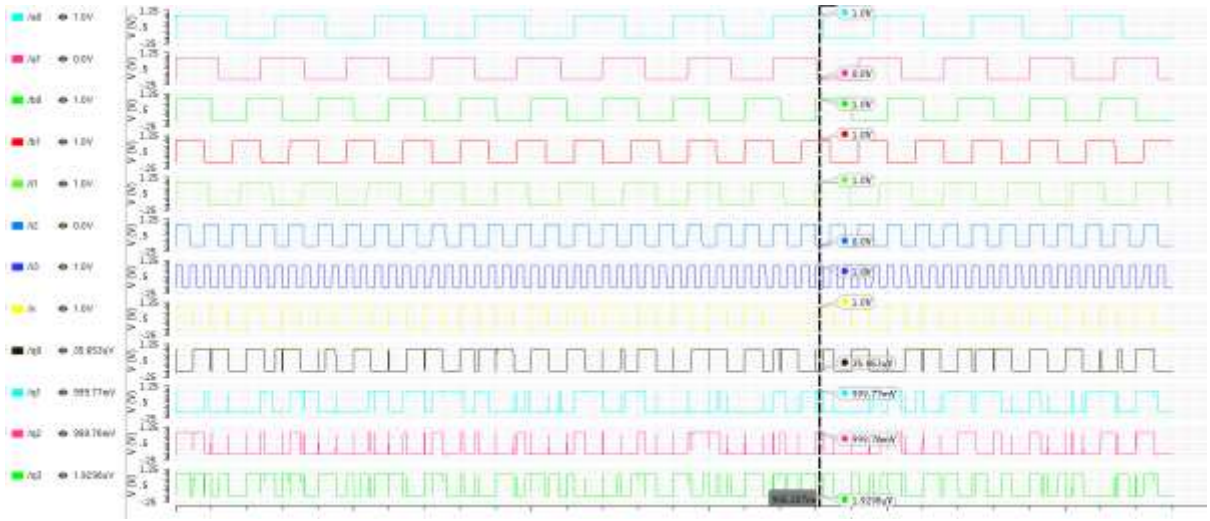

Fig -19 Output waveform of Optimized 2x2 Vedic multiplier

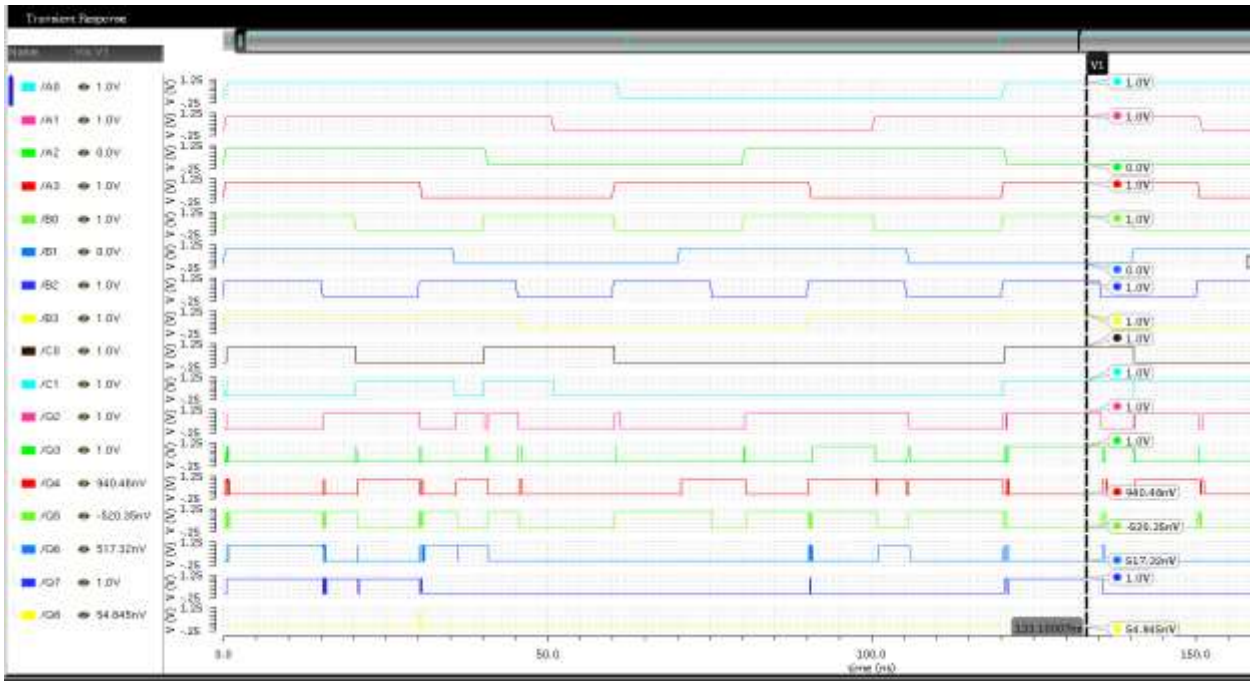

Fig -20 Output waveform of Non-Optimized 4x4Vedic multiplier

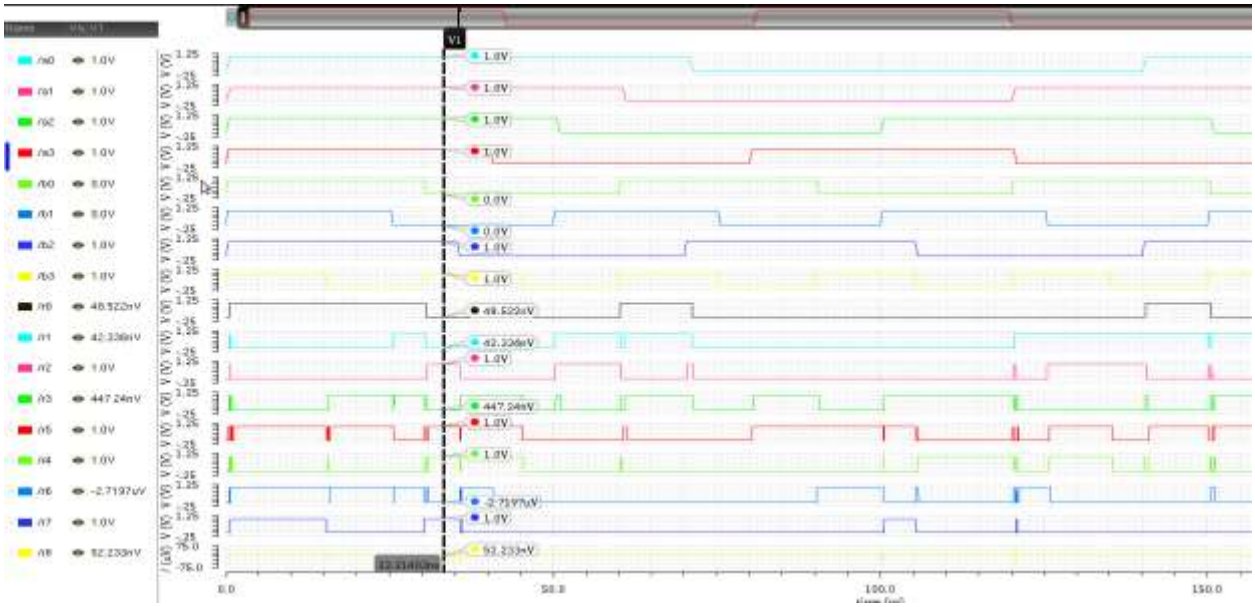

Fig -21 Output waveform of Optimized 4x4Vedic multiplier

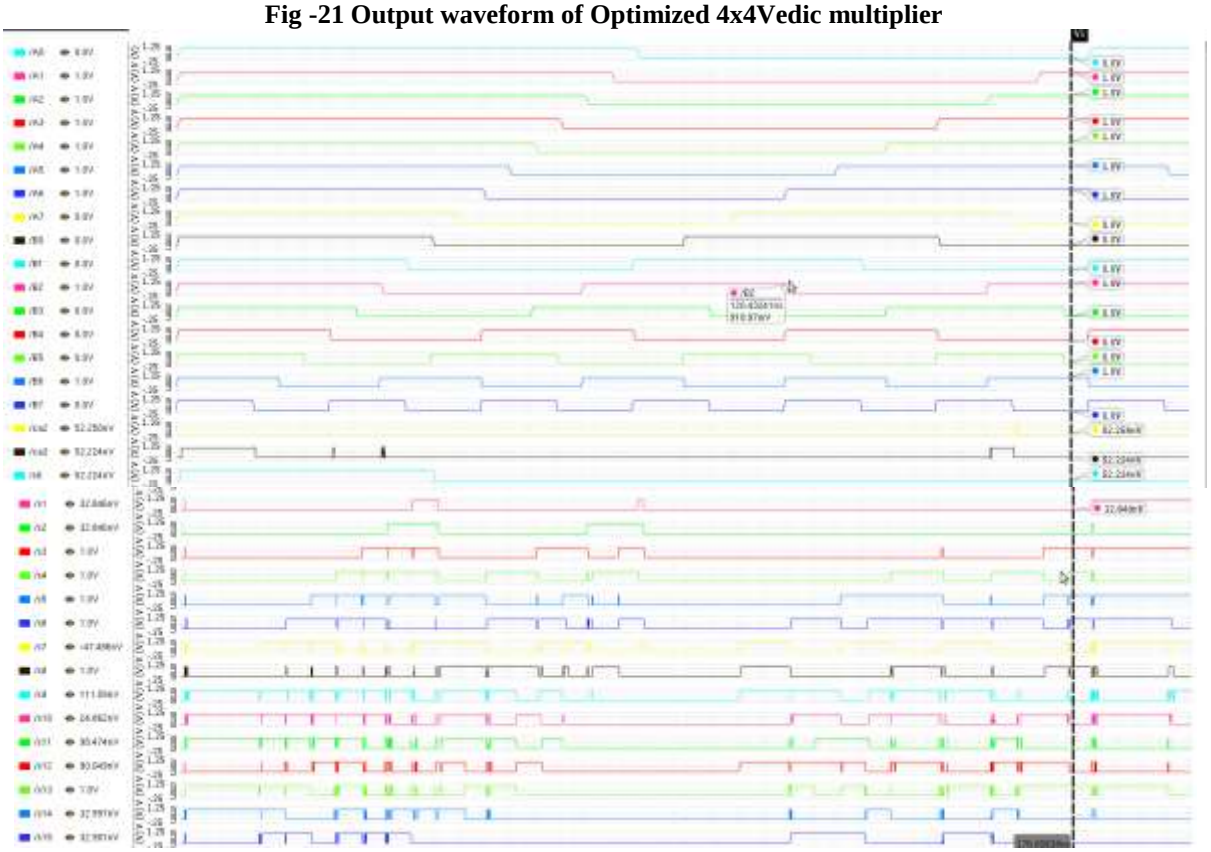

Fig -22 Output waveform of 8x8 Vedic multiplier

Table of Results

Sl.no	Circuit	Power for 90nm	Power for 45nm	Delay for 90nm	Delay for 45nm
1	Inverter	68.03 nW	4.55 nW	10.46 ps	63.13ps
2	AND Gate	107.6 nW	21.89 nW	30.31 ns	153.1 p
3	OR Gate	84.41 nW	17.66 nW	50.795 ns	116.7 ps
4	XOR Gate	321.5 nW	36.97 nW	80.79 ps	47.74 ps
5	Buffer Gate	438.6 pW	159.8 pW	20.2 ns	26.25 ps
6	Feynman Gate	268.7 nW	37.04 nW	40.21 ns	57.03 ps
7	Peres Gate	791.8 nW	116.4 nW	13.58 ns	64.79 ps
8	BVPPG Gate	784.6 nW	125.1 nW	33.87 ns	56.112 ps
9	Binary Vedic Multiplier	1.236 µW	0.263 µW	10.277 ns	293.06 ps
10	Non optimized 2x2 Vedic multiplier	10.24 µW	0.558 µW	21.05 ns	281.64 ps
11	Optimized 2x2 Vedic multiplier	2.46 µW	0.482 µW	29.22 ns	278.06 ps
12	Nonoptimized 4X4 Vedic multiplier	23.19 µW	5.004 µW	573.21 ps	360.02 ps
13	Optimized 4X4 Vedic multiplier	18.3 µW	3.622 µW	480.45 ps	306.86 ps
14	8x8 Vedic multiplier	85.1 µW	9.743 nW	11.53 ns	6.74 ns

6. CONCLUSION

The proposed Vedic multiplier proves to be highly efficient in terms of speed, power, and delay. In normal Vedic multiplier, the power dissipation is more compared to the Vedic multiplier using reversible gates. Hence Vedic multiplier based on the Urdhva Tiryakbhayam algorithms provides the best results in terms of delay and power. In both nonoptimized and optimized 2x2, 4x4 and 8x8 Vedic multiplier using 90nm and 45nm technology both power and delay has been optimized using reversible gates as shown in the above table(1). Hence a number of computations in multiplier using Urdhva Tiryakbhayam logic are less compared to the normal binary multiplier.

7. REFERENCES

[1] Swami Bharati Krsna Tirtha, Vedic
[2] Rakshith T.R. "Optimized Reversible Vedic Multipliers for High-Speed Low Power Operations". *IEEE Conference on Information and Communication Technologies 2013.*
[3] Prof.Rameshwar Rao " Implementation Of Vedic Multiplier using Reversible Gates". *Fifth International Conference on Advances in Computing and Information Technology CSIT 2015.*
[4] J.Radhika, Gutha Viswanath "An 8X8 Design for Urdhva Tiryakbhayam Multiplier using Reversible Logic Gates". *International journal of scientific engineering and technology research 2014.*
[5] Vedic Mathematics: *http://www.hinduism.co.za/vedic.html*

