8X8 VEDIC MULTIPLIER USING REVERSIBLE LOGIC GATE

Rehan.S
[1]P.G. Student, Department of Electronics Engineering., Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune-43, EmailId :rrshaikhpg21-elect@bvucoep.edu.in

Dr.S.K.Oza
[2] Professor and Head, Department of Electronics and Telecommunication Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune-43.
Email Id :skoza@bvucoep.edu.in

ABSTRACT

As multiplication needs more iterations, a longer processing time, and a larger system footprint than other computations, it is the multiplier that has the most impact on the performance of DSP applications. Hence, a high-speed, low-power multiplier is necessary to enhance the system's performance. The ancient discipline of mathematics known as Vedic Mathematics uses a special method of computation based on 16 Sutras. It employs the Urdhva Triyagbhyam Vedic technique, one of the 16 sutras. The Urdhva Triyagbhyam algorithm (sutra) of Vedic mathematics is used for multiplication to increase the speed, power, and area of multipliers. It is implemented using reversible logic gates. Reversible logic is extremely effective at reducing power loss. In this paper, literature survey for Vedic Multiplier is described. A detailed study on Various reversible logic gates used for Vedic Multiplication is presented here. 8 X 8 Vedic multiplication using Han-Carlson adder is shown here using an example.

KEYWORDS: Han-Carlson Adder, Reversible logic gates, Vedic Multiplier, Urdhva Triyagbhyam Sutra.

I.INTRODUCTION

The multipliers are crucial elements for applications like digital signal processing and many more. With technological advancements, it is vital to create multipliers with high speed, low power consumption, small footprints, or perhaps a combination of all three. Addition, subtraction, and shifting operations are used in the conventional approach of multiplication. The partial product that is produced after each computation step is what mostly affects how well the multiplier works. The operation takes longer to complete because of the repetitive adding.

Booth's Algorithm was created to multiply both signed and unsigned numbers in order to decrease the number of rounds. It starts with the ability to add and subtract, and there are numerous ways to compute a product. The two's complement notation of signed binary values is used for multiplication in the Booth's method. The delay is reduced since this multiplier can scan three bits simultaneously. However, because this multiplier uses more power, the system's efficiency declines [1,3]. The computation time for the Booth algorithm multiplier is longer than for the Vedic multiplier. Compared to the Booth multiplier, the Vedic multiplier uses the least amount of hardware space.

Since the Booth multiplier needs more adder cells than the Vedic multiplier, more space is used and more energy is used. The Vedic multiplier is faster and uses less energy than the traditional Booth multiplier. The Vedic multiplier is a good way to lower the system's power usage [1,3].

Sixteen Vedic sutra make up the Vedic multiplier, which is designed using the "Urdhva Triyagbhyam sutra" from these sixteen sutra. Reversible logic gates are used to develop the method that is being suggested. This multiplier can be used in applications like FFT, FIR filters, and several image processing techniques [1].

In section-II, Urdhva Triyagbhyam Sutra is described. Basic of reversible logic gate is shown in Section-III. Han-Carlson Adder is shown in Section-IV. An example of 8 X 8 Vedic multiplication is shown in Section V. Conclusion is described in Section VI.

II. URDHVA TRIYAGBHYAM SUTRA

The Urdhva Triyagbhyam Sutra is the most effective sutra (algorithm), causing the least amount of delay when multiplying any kind of number, no matter how big or tiny. Three phases make to the parallel architecture of the Vedic multiplier. The 4 bit Vedic Multiplication unit makes up the first stage. The second stage consists of carry and partial products. The outcome of the multiplication is combined with adders in the third stage. Here, ripple carry adders are employed as the adders. The Reversible Gates were used in the design of the Ripple CarryAdder.

It illustrates the effectiveness of the Urdhva Triyagbhyam Vedic method for multiplication, which makes a distinction in the multiplication procedure itself. It removes superfluous multiplication steps and allows for the concurrent production of partial products. A hardware description language called Verilog is used to model the proposed method. As a result, we will use reversible logic gates to construct the Urdhva Tiryagbhyam algorithm in this, thus addressing crucial issues such multiplier implementation speed and power consumption. The 8x8 Vedic multiplier will have its architecture optimised in this work by fewer logic gates, constant inputs, and garbage outputs. Several domains, including convolution, filter applications, cryptography, and communication, can benefit from the use of this multiplier [10].

III. BASICS OF REVERSIBLE LOGIC GATE

One of the main challenges with modern design is power dissipation. The most of the designed multiplier uses reversible logic to cut down the power consumption. If the (Boolean) function that computes a gate is bijective then the gate is reversible. The smallest possible number of reversible logic gates should be used while designing reversible circuits. There are various factors that can be taken into account while designing reversible circuits in order to gauge their complexity and effectiveness.

A reversible circuit has the same number of input wires and output wires as a reversible gate. Reversible circuits are depicted as collections of horizontal lines that stand in for the wires. Symbols with a vertical axis are used to depict gates[2][3][7]. As an illustration, the accompanying Figure 3.1 shows a reversible circuit created using Feynman's new notation. It is clear from the truth table shown in Table 3.1 that it is a reversible gate.

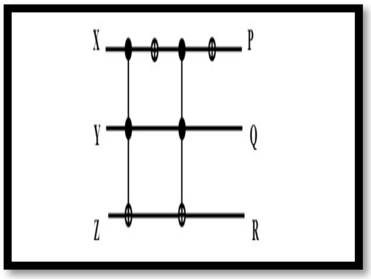
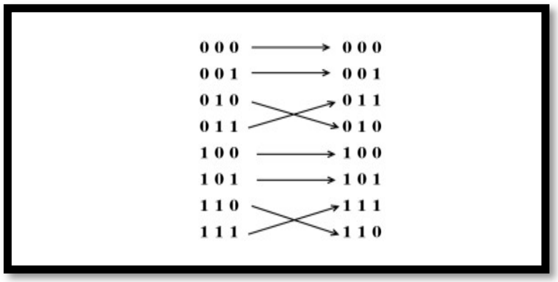


Figure.3.1: Feynman gate [2]

Table 3.1: Multiplied table of Feynman gate[2]



0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	0 1 1
0 1 1	0 1 0
1 0 0	1 0 0
1 0 1	1 0 1
1 1 0	1 1 1
1 1 1	1 1 0

A controlled NOT is another name for the 2*2 gate known as the Feynman gate. For fan-out purposes, it is frequently utilised. P=A, Q=A XOR B are the outputs from the inputs (A,B). Q equals B when A = 0 and B when A = 1, respectively. By using only 2*2 Feynman gates and inverters, one can construct any linear reversible function. Feynman gate is employed as a fan-out gate when B=0[2].

Mapping between the input and output vectors is shown in the table 3.1. The output is balanced as well.

A. TYPES OF REVERSIBLE LOGIC GATES

1. Toffoli Gate.
2. Peres Gate.
3.URG Gate.
4. BJN Gate.
5. HNG Gate.

A.1 TOFFOLI GATE

Table 3.2: Truth table of Toffoli gate[3]

INPUTS			OUTPUTS		
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

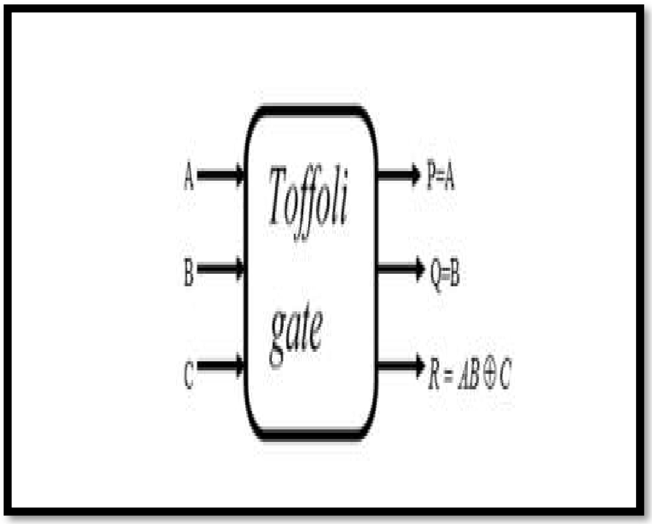


Figure 3.2 : Block diagram of Toffoli gate[3]

Figure 3.2 shows Toffoli gate, which is a 3x3 gate. The output vector is O, and the input vector is I (A, B, and C). The outputs are defined as P=A, Q=B, R=AB xor C. A Toffoli gate has a quantum cost of 5. The Toffoli gate, also known as the CCNOT gate, was developed by Tommaso Toffoli and is a universal reversible logic gate in logic circuits. This means that Toffoli gates can be used to build any reversible circuit. The name "controlled-controlled-not" gate also refers to this gate's function. It features 3-bit inputs and outputs. The third bit is inverted if the first two bits are set; otherwise, all bits remain unchanged [5].

Table.3.3 : Truth table of Peres gate[7]

INPUTS			OUTPUTS		
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

A.2 PERES GATE

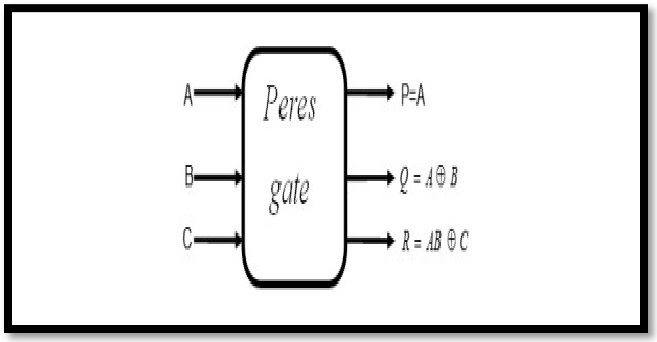


Figure 3.3: Block diagram of Peres gate[7]

Figure 3.3 shows Peres gate where the output vector is O, and the input vector is I (A, B, and C) (P, Q, R). P = A, Q = A xor B, and R=AB xor C define the output. A Peres gate has a quantum cost of 4. Because it has the lowest quantum cost, the Peres gate is utilised in the suggested design[7]

A.3 URG GATE

Table 3.4.: Truth table of URG gate[7]

INPUTS			OUTPUTS		
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	0	1	1
1	1	1	0	1	0

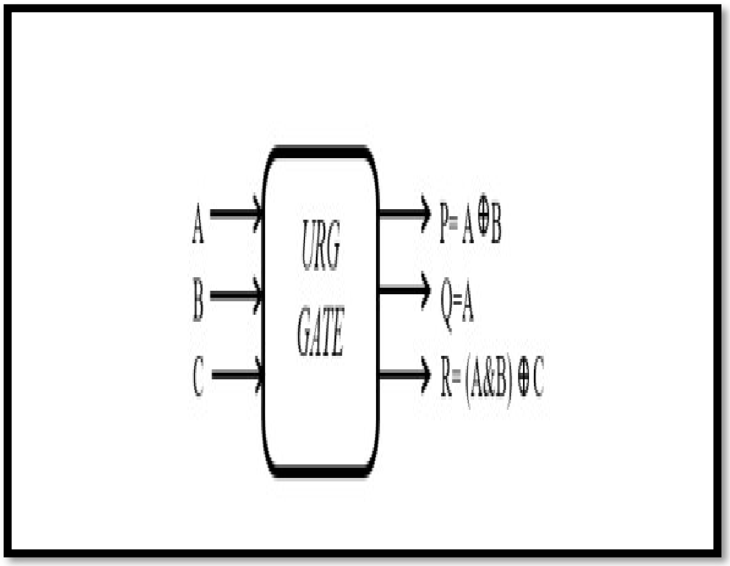


Figure 3.4: Block diagram of URG gate[7]

Figure 3.4 shows URG gate which is a 3*3 gate with inputs (A, B, C) and outputs P= (A+B) XOR C, Q= B, R = ABC.

A.4 BJN GATE

Table 3.5 : Truth table of BJN gate[7]

INPUTS			OUTPUTS		
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	0	0
1	0	0	1	1	1

1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	1	1	0

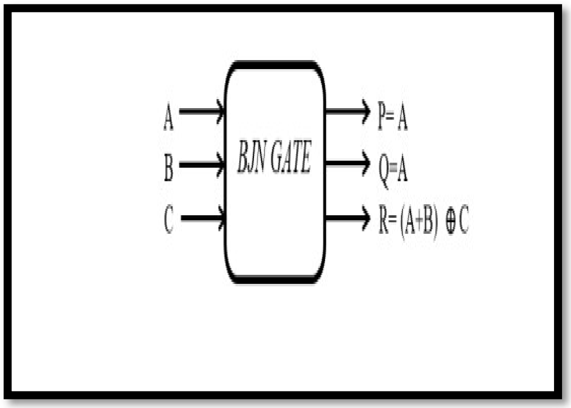


Figure 3.5: Block diagram of BJN gate[7]

Figure 3.5 shows the logic diagram and truth table of the proposed new Reversible BJN BJN Gate which is a 3*3 gate with inputs (A, B, C) and outputs P=A, Q=B, R =(A+B) xor C. It has quantum cost of 5.

A.5 HNG GATE

Table 3.6 : Truth table of HNG gate[3

INPUTS				OUTPUTS			
A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0

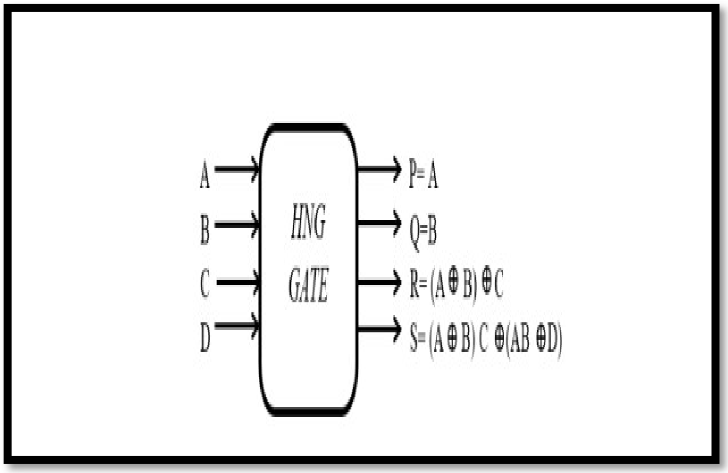


Figure 3.6 : Block diagram of HNG gate[3]

Figure 3.4 shows HNG gate which is a 4*4 gate with inputs (A, B, C,D) and outputs P=A, Q=B, R= (A XOR B)XOR C, S= (A XOR B) C XOR(AB XOR D) (AB XOR D).

IV. HAN – CARLSON ADDER

The section III described various reversible logic gates which were used in existing Vedic multiplier design. However, in most of the work, the implemented design uses ripple carry adders made of reversible logic gates in which area, power consumption and delay performance were not satisfactory. Hence, Han-Carlson adder can be used to overcome these limitations.

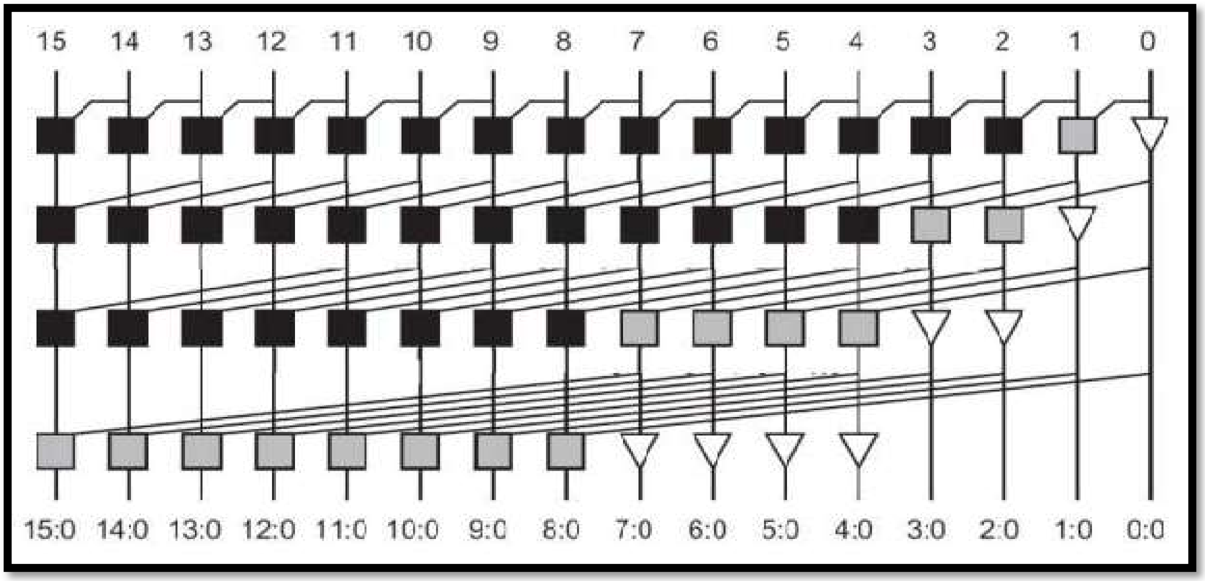


Figure no 4.1 : Han-Carlson Adder (Mix of Kogge-Stone and Brent-Kung)[5]

Figure 4.1 shows Han-Carlson Adder, which is made from the mix of Kogge – Stone and Brent – Kung. The Kogge-Stone adder is a fast adder circuit that uses a tree structure to perform addition. It has a better worst-case delay compared to the Han-Carlson adder, but requires more circuitry.

The Brent-Kung adder is another type of fast adder circuit that uses a similar tree structure to the Kogge-Stone adder, but with a more efficient carry propagation scheme. It has a better area-delay product compared to the Kogge-Stone adder.

One way to combine these adder circuits is to use a hybrid structure that mixes the Han-Carlson, Kogge-Stone, and Brent-Kung adders together. For example, we could use the Han-Carlson adder for the first few stages of the adder, and then switch to the Kogge-Stone or Brent-Kung adder for the later stages.

Han-Carlson is a tree of a family of networks between Kogge-Stone and Brent-Kung adders. The outer rows of this adder perform Brent-Kung addition methodology while the inner rows additions are performed by KoggeStone methodology. Han-Carlson adder uses Brent-Kung at the beginning and at the end of the prefix graph and the number logic level is defined as [log2 (n) + 1] where „n" represents the number of bits. The black and grey cells are placed at the odd bit positions in the initial stage and the middle stages. Only grey cells are placed at the even bit positions in the final stage where carry merge operation happens. The delay of this structure is [log2 (n) +1] and the hardware complexity is given by [(n/2) log2 (n)]. Han-Carlson gives good trade-off between fanout, number of logic cells and the number of black cells compared to Kogge-Stone adder.

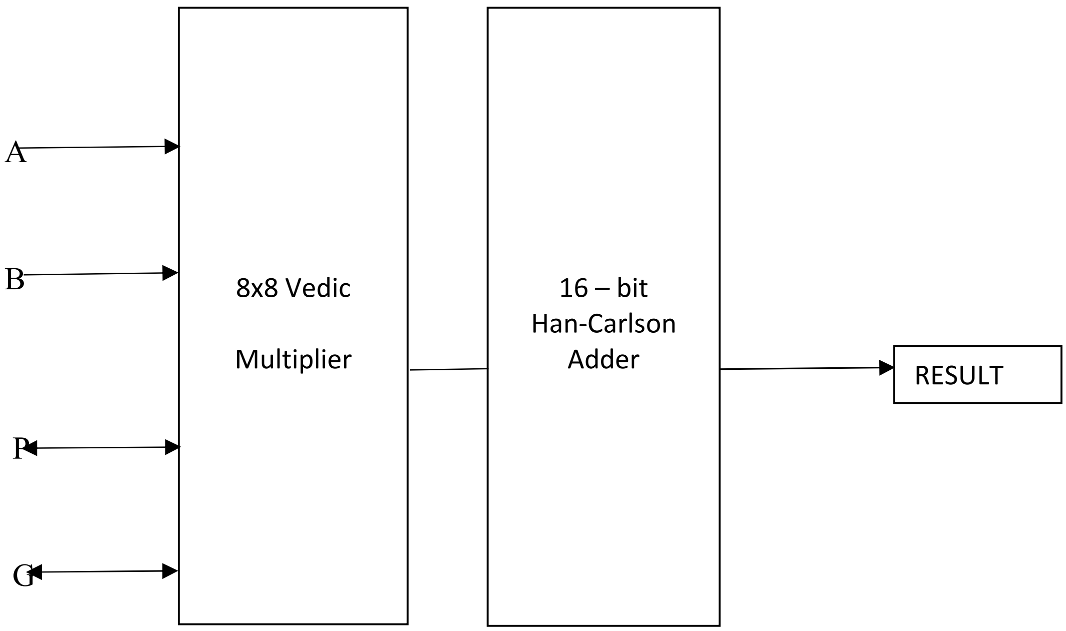


Figure 4.1 : 8X8 Vedic Multiplier using 16-bit Han-Carlson Adder

In Figure 4.1 shows the 8x8 Vedic Multiplicaton using 16-Bit Han-Carlson Adder where A and B are the two 8-bit numbers to be multiplied, and P and G are the partial products and generates generated by the Vedic multiplier.

The Vedic multiplier decomposes the two numbers A and B into smaller parts, performs multiplication using cross products, and generates the partial products and generates P and G.

The Han-Carlson adder then adds up the partial products P using carry-lookahead logic to reduce the delay of the carry signal. The result of the multiplication is obtained at the output of the adder.

Carry lookahead logic overcomes this limitation by computing the carry signals for multiple bits simultaneously, rather than waiting for them to propagate one by one. The idea is to express the carry of each bit as a function of the input bits and the carry signals of the previous bits, without waiting for the actual carry to be generated.

The basic principle of carry lookahead logic is to pre-compute and store intermediate carry signals, called "propagate" and "generate" signals, that indicate whether each bit position generates a carry or just propagates it from the previous bit. These signals can be computed using simple logic gates based on the input bits, and can be combined using more complex logic to generate the final carry signals for the entire adder.

The advantage of carry lookahead logic is that it allows the carry signals to be computed in parallel, rather than sequentially, which reduces the delay of the carry path and improves the overall performance of the adder. However, it requires additional logic gates and complexity, which may increase the area and power consumption of the circuit.

V. 8 X 8 Vedic Multiplication

To perform 8x8 Vedic multiplication using a 16-bit Han-Carlson adder the following steps can be carried out:

Decompose the two 8-bit numbers into two parts of 4 bits each. For example, let's say we want to multiply A = 11010110 and B = 10011011. We can decompose them as follows:

A = 1101 0110
B = 1001 1011

Multiply the left parts of A and B and the right parts of A and B separately using standard multiplication. For example, we get:

A left x B left = 1101 x 1001 = 11000101
A left x B right = 1101 x 1011 = 11101011
A right x B left = 0110 x 1001 = 01010110
A right x B right = 0110 x 1011 = 01011110

Add the cross-products using a 16-bit Han-Carlson adder. We get:

1 1000 0100 1111 0010

Note: The first bit is the carry bit, which is ignored in the result.

Shift the results by the appropriate number of bits and add them together to get the final result. Specifically, we shift A left x B left by 8 bits to the left, A left x B right and A right x B left by 4 bits to the left, and then add all the shifted results together. We get:

101101100110110.

VII. CONCLUSION

High speed multipliers are required in many digital signal processing and image processing applications. In this paper, a literature survey for Vedic Multiplier is described. A detailed study on various reversible logic gates used for Vedic Multiplication is presented here. Han-Carlson adder can perform better in terms of area, power and delay. A high-performance Vedic multiplier using Han-Carlson adder is shown with example of 8 X 8 multiplication. In future focus, the shown 8 X 8 Vedic multiplier can be implemented on hardware using Spartan and LCD.

REFRENCES

1. M. Mohamed Asan Basiri and Sk. Noor Mahammad, “Configurable Folded IIR Filter Design”, IEEE Journal of Transactions on Circuits and Systems - II: Express Briefs, Vol. 62, No. 12, pp. 1144 – 1148, 2019.

2. Darjn Esposito, Davide De Caro, Senior Member,IEEE, Ettore Napoli, Nicola Petra, Member, IEEE and Antonio Giuseppe Maria Strollo, Senior Member, IEEE, “International Journal of Engineering And Advanced Technology”, “ Variable Latency Speculative Han-Carlson Adder”, May 2021.

3. Hemangi Patil,S.Dsawant” Design and Implementation of Energy Efficient Vedic Multiplier using FPGA” 2019 International Conference on Information Processing Vishwakarma Institute of Technology. Dec 18-19, 2019.

4. B. Parameswaran, K. R. Venugopal, and L. M. Patnaik, "Design and implementation of an 8x8 Vedic multiplier using reversible logic gates," Journal of Circuits, Systems and Computers, vol. 23, no. 5, 2014.

5. Swathi.U.” FPGA Implementation of Optimized Reversible Vedic Multiplier Architectures For High Speed Low Power Applications.” International Advanced Research Journal in Science, Engineering And Technology Vol. 7, Issue 3, March 2020.

6. C.Devika, Dr.Baswaraj, Zameerahamad” Optimized 64 Bit Mac Using Vedic Multiplier And Reversible Logic For Dsp Application” Asia Journal Of Engineering & Technological Perspective Research (Asia-Jetpr) Volume 2; Issue 3; May-June-2020 ; Page No. 10-15.

7. Venkata Latha, Syamalay. Anilchowdary. T,Muralikrushna G” Reversible Vedic Multiplier” International Journal of Engineering And Advanced Technology, Volume-8, Issue-6s2, August 2019.

8. Venkata Latha,Syamalay.Anilchowdary.T,Muralikrushna G” Reversible Vedic Multiplier” International Journal of Engineering And Advanced Technology, Volume-8, Issue-6s2, August 2019.

9. R.P.Brent and H.T.Kung, „“A regular layout of parallel adders”,IEEE , “Journal of Transaction of Computers” , Vol C-31, Mar 2020 .

10. P.M.Kogge and H.S.Stone, "A Parallel Algorithm for the efficient solution of a general class of recurrence equations", IEEE "Journal of Transaction of.computers" , vol C-22, Aug 2021.

11. L. Sriraman, and T. N. Prabakar, "Design and Implementation of Two Variable Multiplier Using KCM and Vedic Mathematics," IEEE, "International Journal of Engineeirng Research and Technology", 2012.

12. Konstantinos Vitoroulis, and Asim J. Al-Khalili, "Performance of Parallel Prefix Adders implemented with FPGA technology," NEWCAS 2007. IEEE Northeast Workshop on Circuits and Systems, 2007., pp.498 – 501, 5-8 Aug. 2007.