International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

DESIGN AND EVALUATION OF AN ENERGY-EFFICIENT HIGH-SPEED MULTIPLIER USING A QUATERNARY CARRY LOOK-AHEAD ADDER

Sadia Anjum[1], Dr. M. Pavithra Jyothi[2]
[1]PG Scholar, Department of ECE, Shadan Women's College of Engineering and Technology, Hyderabad, sadiaanjum5555@gmail.com
[2]Professor, Department of ECE, Shadan Women's College of Engineering and Technology m.pavithrajyothi@gmail.com

ABSTRACT

Need of Digital Signal Processing (DSP) systems which is embedded and portable has been increasing as a result of the speed growth of semiconductor technology. Multiplier is a most crucial part in almost every DSP application. So, the low power, high speed multipliers is needed for high-speed DSP applications. Vedic multiplier is one of the fastest and efficient multipliers, Main algorithm of Vedic multiplication is Urdhva Triyakbhyam. It is a general multiplication formula applicable to all cases of multiplication. It literally means Vertically and Crosswise.

The multiplication is depending upon the previous computations of partial sum to produce the final output, so we need to design efficient adders. In arithmetic operation, major issue corresponds to carry in binary number system. Higher radix number system like Quaternary Signed Digit (QSD) can be used for performing arithmetic operations without carry. Designing the adder using QSD number representation allows fast addition which is capable of carry free addition because the carry propagation chain is eliminated, hence it reduces the propagation time in comparison with radix 2 system.

In this project, a Vedic multiplier using Quaternary Carry Look Ahead Adder (QCLA) is existed design and Vedic multiplier using Quaternary Carry Increment Adder (QCIA) is proposed design, it has less area compared with the Existed design. In this project Xilinx-ISE 14.7 tool is used for simulation, logical verification, and further synthesizing and the HDL language used for the project is VERILOG.

INTRODUCTION

1.1 Multiplier

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

The common multiplication method is "add and shift" algorithm. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Vedic multiplier using carry look ahead adder is one of the most popular Urdhva Tiryakbhayam method. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics.

The basic method of multiplier is explaining below

```
   123
 x 456
 =====
   738  (this is 123 x 6)
  615   (this is 123 x 5, shifted one position to the left)
+ 492    (this is 123 x 4, shifted two positions to the left)
 =====
 56088
```

1.2 Binary Multiplier

The binary multiplication also happens in same way of digit multiplication as shown in below example here by getting partial products and gates are used and we are using adder (half adder, full adder) adding the columns.

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

        1 0 1 0
      0 0 0 0
    1 0 1 0
  ---------------------

  1 1 0 0 1 0
```

An example of 4-bit multiplication method is shown below:

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

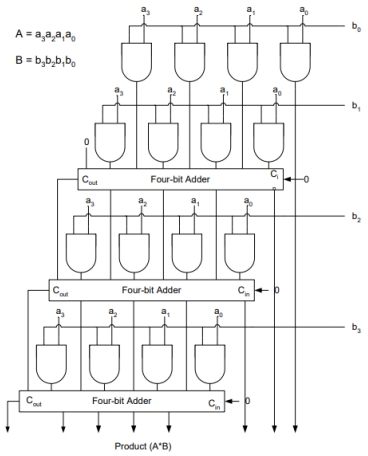
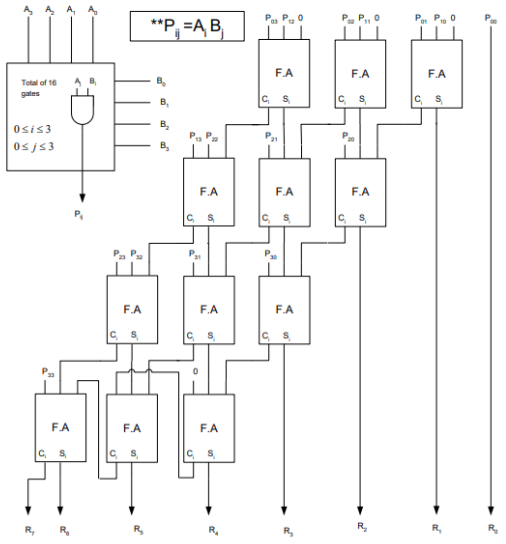


Fig 1.1: 4-bit multiplier

1.3 Array Multiplier

Although the method is simple as it can be seen from this example, the addition is done serially as well as in parallel. To improve on the delay and area the CRAs are replaced with Carry Save Adders, in which every carry and sum signal is passed to the adders of the next stage. Final product is obtained in a final adder by any fast adder (usually carry ripple adder). In array multiplication we need to add, as many partial products as there are multiplier bits. This arrangement is shown in the figure below



Total Area = (N-1) * M * Area $_{FA}$
Delay= 2(M-1) τ_{FA}

Fig:1.2 Array multiplier

In applications like multimedia signal processing and data mining which can tolerate error, exact computing units are not always necessary. They can be replaced with their approximate counterparts. Research on approximate computing for error tolerant applications is on the rise. Adders and multipliers form the key components in these applications. In, approximate full adders are proposed at transistor level and they are utilized in digital signal processing applications.

1.4 Implementation of Wallace Multiplier

The Wallace tree has three steps:

1. Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding results. Depending on position of the multiplied bits, the wires carry different weights, for example wire of bit carrying result is 32.
2. Reduce the number of partial products to two by layers of full and half adders.
3. Group the wires in two numbers, and add them with a conventional adder.
4. The second phase works as long as there are three or more wires with the same weight add a following layer:

- Take any three wires with the same weights and input them into a full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.
- If there are two wires of the same weight left, input them into a half adder.
- If there is just one wire left, connect it to the next layer.

a) Steps involved in WALLACE TREE multipliers Algorithm:

- ➲ Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding N results. Depending on position of the multiplied bits, the wires carry different weights.
- ➲ Reduce the number of partial products to two layers of full adders.
- ➲ Group the wires in two numbers, and add them with a conventional adder.

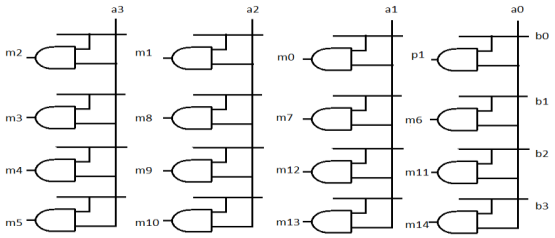


Fig:1.3 Product terms generated by a collection of AND gates

b) WALLACE TREE Multiplier Using Adder

Ripple Carry Adder is the method used to add a greater number of additions to be performed with the carry in sand carry outs that is to be chained. Thus, multiple adders are used in ripple carry adder. It is possible to create a logical circuit using several full adders to add multiple-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is a ripple carry adder, since each carry bit "ripples" to the next full adder. The proposed architecture of WALLACE multiplier algorithm using RCA is shown in Fig

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

Take any 3 values with the same weights and give them as input into a full adder. The result will be an output wire of the same weight.

- Partial product obtained after multiplication is taken at the first stage. The data are taken with 3 wires and added using adders and the carry of each stage is added with next two data in the same stage.
- Partial products reduced to two layers of full adders with same procedure.

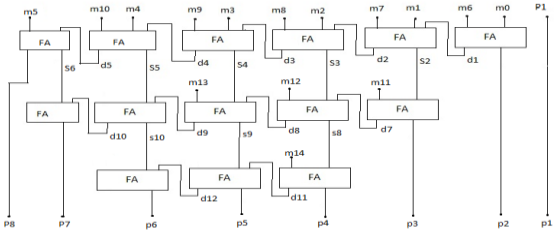


Fig 1.4 : 4x4 Wallace Multiplier

At the final stage, same method of ripple carry adder method is performed and thus product terms p1 to p8 is obtained.

- A fast way to multiply two binary integers.
- Any multiplier has three stages.

Stage1: partial products.
Stage2: partial product addition.
Stage3: final addition.

Stage1 partial products

- Works exactly like "long hand" multiplication. Numbers are binary integers. Products(each blue square) are the result of a simple and gate. All products are done simultaneously. Fast (however long and gates takes).Trick is in adding up the columns

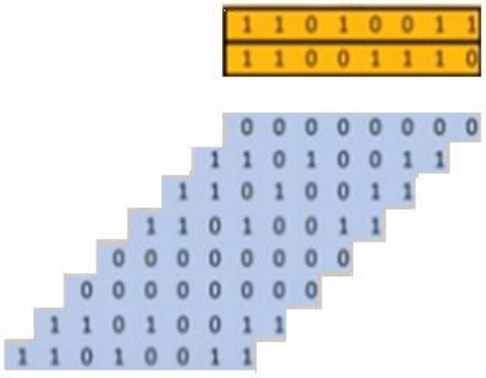


Fig 1.5: **Stage1 partial products**

Stage2: partial product addition, step1

To add up columns, add up three rows at a time. The result for each set of three rows is a set of two rows. Each resulting set of two rows has a row for the sum and a row for the carry-out. Odd rows are left alone.
red: full adder output.
yellow: half adder output.
green: left alone.

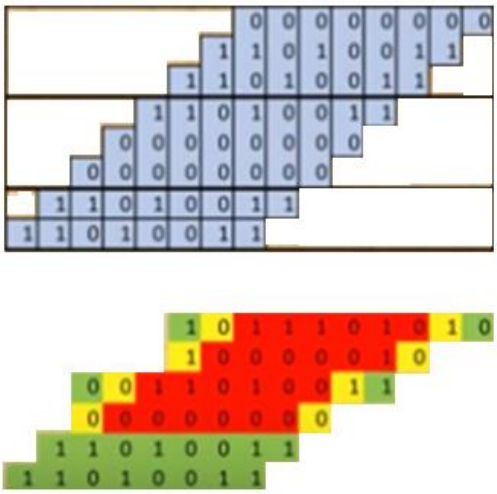


Fig 1.6: partial product addition, step1

Stage2: partial products addition, step2

- Repeat the same process. This time, three are two sets of three rows .Result in two set of two rows . Gray boxes indicate that summation bits have been moved down to the carry_out row.

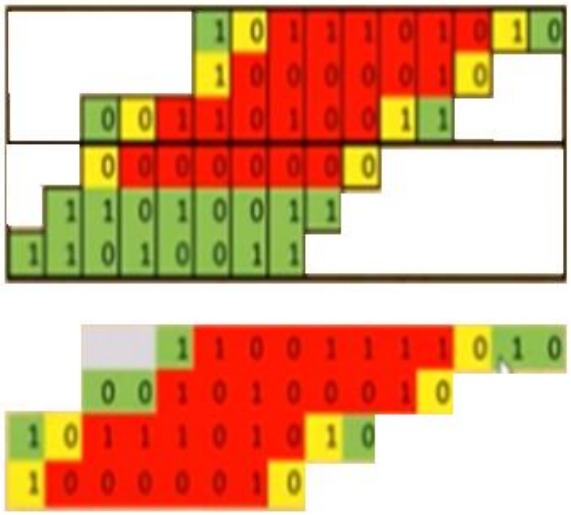


Fig 1.7: partial products addition, step2

Stage2: partial products addition, step4

- Repeat the process one last time. Remaining three rows become two rows. In this example, stage2 has 4 steps, and 4 full adder delays. The five LSB have already been calculated.

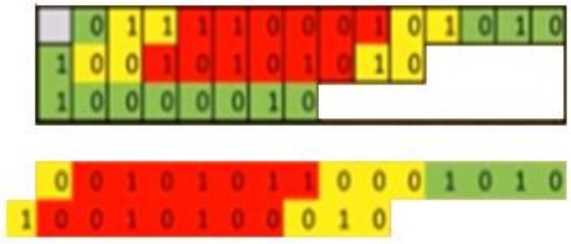


Fig 1.8: **partial products addition, step4**

Stage3: final addition

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

- Final result is calculated by adding the final two rows. In this example, the 5 LSBs do no need to be added. The saving from already having 5bits offsets the delay from doing stage2Result is that Wallace tree multiplication takes about the same amount of time as a 2N _bit ripple carry adder.

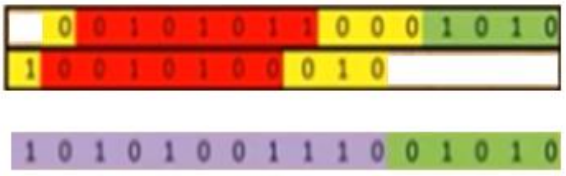


Fig 1.9: **final addition**

In ICs, for implementation of array multiplier need only small space. In digital integrated circuits, it is effective method of multiplication. Shifted partial products accumulation and partial products evaluation are principles used in multiplication. Successive operation of addition is required for performing this operation.

1.5 Adders

Adder is an important component needed for designing multiplier. It may be a Ripple Carry Adder (RCA), Carry Look Ahead (CLA), Carry Select, Carry Skip and Carry Save Adder (CSA). Various fast adder performances are analyzed by several research works. Carry can be computed with less time by carry look-ahead adder. Based on input signals, computed Carry for next stage in Look ahead carry algorithm. So, addition operation can be done very quickly.

1.6 Quaternary Signed Digit (QSD)

In array multiplier architecture, instead of full adders, carry look-ahead adder based on QSD are used. This facilitates low consumption of power and quick multiplication. Addition arithmetic operation defines speed of any processor. The Quaternary Signed Digit number system (QSD) is used by various researchers, in order to enhance the speed of addition operation. Subtraction without borrow and addition without carry are performed by QSD number systems.

When number of bits is increased, carry propagation and formation limits the speed of computation in binary number system. Subtraction without borrow and addition without carry are performed by QSD number systems. Number form −3 to3 are used for representing every digit in QSD. With constant delay, large number of digits like 128,256 and more are implemented in digital implementation. QSD number system has following major advantages. They are reduced interconnections and transistor. Overall complexity of system is reduced by this. This technique can be used for implementing multipliers and adder with effective area and speed.

1.7 QSD logic gates

Binary logic system is used to form a quaternary logic system which is close to binary number system. It is base 4 system. Digits 0, 1, 2 and 3 are used in this system. They are used for representing real numbers. QSD number's signed representation is given by,

$$X = \sum_{i=0}^{n-1} x_i 4^i$$

where, Proper representation of decimal are produced by xi = {3^{-}, 2^{-}, 1^{-}, 0, 1, 2, 3}. QSD positive number's QSD complement produces QSD negative number. For example, 3 = −3, 2 = −2, 1 = −1. With constant delay, large number of digits like 128,256 and more are implemented in digital implementation. QSD number system has following major advantages. They are, reduced interconnections and transistor. Overall complexity of system is reduced by this. This technique can be used for implementing multipliers and adder with effective area and speed.

1.8 Quaternary Inverter

In different logic circuits, an important role is played by Quaternary inverter. Input signal is complemented using inverter function. Table shows Quaternary inverter's truth table.

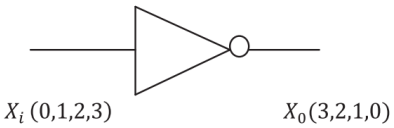


Fig: quaternary inverter

Quaternary inverter truth table.

Input	Output
0	3
1	2
2	1
3	0

Table 1: truth table of quaternary inverter

1.9 Quaternary NAND gate

In different logic circuits, an important role is played by Quaternary NAND gate. Memory circuits are realized using this. Table shows Quaternary NAND gate's truth table. Quaternary NAND gate's output is expressed as,

OUT =~ (ln1.ln2)

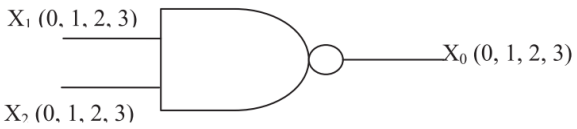


Fig 1.10: quaternary NAND gate

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

In 1 / In 2	0	1	2	3
0	3	3	3	3
1	3	2	2	2
2	3	2	1	1
3	3	2	1	0

Table 2: truth table of quaternary NAND gate

1.10 Quaternary NOR gate

Memory circuits like flip-flops are realized using Quaternary NOR gate symbol Implementations of counters, shift registers are done using this flip-flop. Quaternary NOR gate's output is defined as,

OUT =~ (ln1 + ln2)

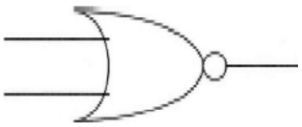


Fig: quaternary NOR gate

In 1 / In 2	0	1	2	3
0	3	2	1	0
1	2	2	1	0
2	1	1	1	0
3	0	0	0	0

Table 3: truth table of quaternary NOR gate

LITERATURE SURVEY

S. Mukherjee, D.K. Soni, Energy efficient multiplier for highspeed DSP application, Int. J. Comput. Sci. Mob. Comput

A multiplier is one of the important hardware blocks in most digital and high-performance systems such as microprocessors. With improving in technology, many researchers have tried and are trying to design multipliers which offer high speed, low power consumption and less area. However, area and speed are two most important constraints. In this paper we propose Energy Efficient approximate multiplier using AHA and AFA. Due to this area reduces up to 35% and carry propagation delay also reduces.

M. Sutha, Dr. R. Nirmala, Dr. PT. Kalaivani, Design of Low Power Less Leakage Quaternary Adder Using Finfet

The Multi-valued logic (MVL) was more efficient logic than binary logic. A QTL Full Adder (QFA) is designed with support of Fin-Field Effect Transistor (FinFET) technology. FinFET provides low static leakage current, low switching voltage and power consumption. The Main aim is to design a Quaternary adder with minimum power and leakage with FinFET Technology.

M. Bala Murugesh, S. Nagaraj, J. Jayasree, Modified High Speed 32-bit Vedic Multiplier Design and Implementation,

The proposed research work specifies the modified version of binary Vedic multiplier using Vedic sutras of ancient Vedic mathematics. It provides modification in preliminarily implemented Vedic multiplier. The modified binary Vedic multiplier is preferable has shown improvement in the terms of the time delay and also device utilization. The proposed technique was designed and implemented in Verilog HDL. For HDL simulation, model sim tool is used and for circuit synthesis, Xilinx is used. The simulation has been done for 4-bit, 8-bit, 16-bit, 32-bit multiplication operation. Only for 32-bit binary Vedic multiplier technique the simulation results**, Design and analysis of an array multiplier using an area efficient full adder cell in 32 nm CMOS technology,**

Multiplier is one of the basic functional units in digital signal processor. Most high-performance DSP systems rely on hardware multiplication to achieve high data throughput. In this paper a low power and low area array multiplier is proposed. The conventional array multiplier is synthesized using 16T full adder cell. In conventional array multiplier the final stage of addition is removed and the carry bits are given to the input of the next left column input, thereby causing a large trade off in power and area. The proposed array multiplier is synthesized using 10T full adder cell. The proposed array multiplier design uses 96 less transistor count and saves 2.82% of total power, 13.24% of more speed and 15.69% less power delay product when being compared with the conventional array multiplier in 32nm MOSFET Technology using HSPICE.

D.P. Borkute, P. Patel, P.K. Dakhole , Delay performance and implementation of quaternary logic circuits,

Good Characteristics and advantages of multi-valued logic (MVL) electronic systems and circuits are created great interest for its practical implementation. This paper presents voltage mode quaternary CMOS circuit design using 90nm technology. Basic gates such as quaternary inverter, NMAX, NMIN and Quaternary multiplexer are designed and simulated. Low power consumption of 14 μ W is observed at 2.2GHz with 1.2 V power supply. Circuits are verified using HSPICE simulations. The circuits described here are also suitable to be implemented in classical CMOS VLSI technology.

S. Srikanth, I.T. Banu, G.V. Priya, G. Usha, Low power array multiplier using modified full adder,

Designing multipliers that are of high-speed, low power, and regular in layout are of substantial research interest. Speed of the multiplier can be increased by reducing the generated partial products. Many attempts have been made to reduce the number of partial products generated in a multiplication process one of them is array multiplier. array multiplier half adder has

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

been used to sum the carry products in reduced time. Achieving high speed integrated circuits with low power consumption is a major concern for the VLSI circuit designers. Most arithmetic operations are done using multiplier, which is the major power consuming element in the digital circuits. Basically, the process of multiplication is realized in hardware in terms of shift and add operation. The optimization of adder has led to the improvement in performance of multiplier. In this paper, a modified full adder using multiplexer is proposed to achieve low power consumption of multiplier. To analyze the efficiency of proposed design, the conventional array multiplier structure is used. The designs are developed using Verilog HDL and the functionalities are verified through simulation using Xilinx. The ASIC synthesis results of the proposed multiplier shows an average reduction of 35.45% in power consumption, 40.75% in area, and 15.65% in delay compared to the existing approaches.

2.1 VEDIC MEULTIPLIER USING QUATERNARY CARRY LOOKA AHEAD ADDER (EXISTED DESIGN)

4.1.1 QSD adder unit (half adder & full adder)

Common arithmetic circuit corresponds to an adder. In order to realize various arithmetic operations, it is used as a building block. In large adders, mostly used building blocks are full adder and half adder of single bit. Various types of arithmetic circuits also constructed using this.

2.1.2 quaternary half adders (QHA)

Two inputs X and Y are given as input to binary Half-Adder (HA). Carryout bit C and sum output S are produced by this binary Half-Adder (HA). Fig. shows it. Two quaternary digits are given as input for half adder circuit in quaternary logic and quaternary carry and sum digits are produced by this. Single-digit quaternary adder corresponds to a half-adder. For single-digit inputs, it produces two-digit sum as X + Y = (Sum, Carry)4. Table shows truth table of Quaternary Half Adder

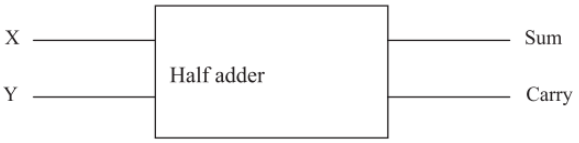


Fig 2.1: quaternary half adder

Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
0	2	2	0
0	3	3	0
1	0	1	0
1	1	2	0
1	2	3	0
1	3	0	1
2	0	2	0
2	1	3	0
2	2	0	1
2	3	1	1
3	0	3	0
3	1	0	1
3	2	1	1
3	3	2	1

Table 4: truth table of quaternary half adder

2.1.3 Quaternary full adder

Quaternary half adder is used for constructing proposed quaternary full adder. Carry-in C_{in}, which is an additional input added in this circuit. Quaternary C_{in} digit and two quaternary digits are added by this circuit. Quaternary carry outputs and sum are generated by this circuit. There will be only two values for Cin pin (0 and 1). Implementation of circuit is simplified by this assumption. Quaternary multipliers and adder implementation are done in a useful manner. Fig. shows quaternary full adder's block diagram. One XOR gate and two half adders are used for constructing Quaternary Full adder. It is like a conventional full adder. Two quaternary inputs X and Y are given to first quaternary half adder. First half adder's sum and another input Cin are given as input to second quaternary half adder. QFA sum is produced by second half adder's sum output. XOR gate's two inputs are connected with carry output of first and second full adder. Proposed full adder's carry output is generated by this output of XOR gate. Table shows truth table of full adder with input carry 0 and Table shows truth table of full adder with carry input 1

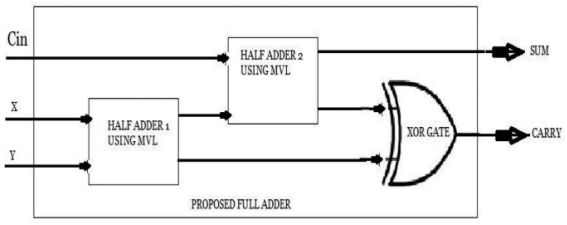


Fig 2.2: quaternary full adder

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

	Inputs		Outputs	
C_{in}	X	Y	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	0	2	2	0
0	0	3	3	0
0	1	0	1	0
0	1	1	2	0
0	1	2	3	0
0	1	3	0	1
0	2	0	2	0
0	2	1	3	0
0	2	2	0	1
0	2	3	1	1
0	3	0	3	0
0	3	1	0	1
0	3	2	1	1
0	3	3	2	1

Table 5: truth table of quaternary full adder cin=0

	Inputs		Outputs	
C_{in}	X	Y	Sum	Carry
1	0	0	1	0
1	0	1	2	0
1	0	2	3	1
1	0	3	0	0
1	1	0	2	0
1	1	1	3	0
1	1	2	0	1
1	1	3	1	1
1	2	0	3	0
1	2	1	0	1
1	2	2	1	1
1	2	3	2	1
1	3	0	0	1
1	3	1	1	1
1	3	2	2	1
1	3	3	3	1

Table 6: truth table of quaternary full adder cin=1

2.1.4 Quaternary carry look ahead adder

In architecture of adder, Carry-Lookahead Adder (CLA) architecture is commonly used. Carry generate and propagate logic functions are used in this. Better performance is exhibited by this. Fig. shows this adder. Computation of carry bit duration is minimized, which enhances the speed of CLA.

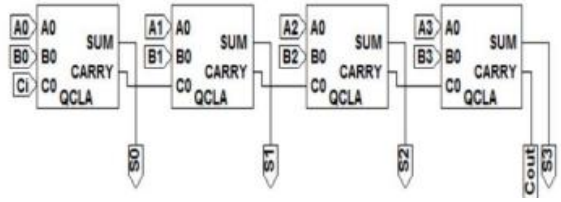


Fig: quaternary carry look ahead adder using QFA

A Quaternary 4 BIT CARRY LOOK AHEAD is designed using Quaternary full adder.

Here one quaternary full adder is worked as carry look ahead adder

2.1.5 Vedic multiplier

Vedic Mathematics is one of the most ancient methodologies used by the Aryans in order to perform mathematical calculations. This consists of algorithms that can boil down large arithmetic operations to simple mind calculations. The above said advantage stems from the fact that Vedic mathematics approach is totally different and considered very close to the way a human mind works. The Urdhva Tiryakbhayam is one such multiplication algorithm which is well known for its efficiency in reducing the calculations involved. The mode used by Vedic multiplier is Vedic mathematics. By using this technique, it will increase, and consumes fewer hardware elements. The sutra used by Vedic multiplier is Urdhva Tiryakbhyam which means Vertically as well as Crosswise.

2.1.6 Vedic multiplier using QCLA

The Vedic multiplication methodology is done in the following for 4-bit inputs, A(A3 -A0) and B(B3 -B0) and 8-bit output S (S7 -S0)

$A_3A_2A_1A_0$
x $B_3B_2B_1B_0$

$(A_3A_2)x(B_1B_0)$ $(A_1A_0)x(B_1B_0)$
$(A_3A_2)x(B_3B_2)$ $(A_1A_0)x(B_3B_2)$

S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0

A 2-bit multiplier of figure shown below is used to calculate intermediate stage results, and the output is 4 bits. (A3A2) (B3B2) using 2-bit multiplier generates result: S33 S32 S31 S30
(A3A2) (B1B0) using 2-bit multiplier generates result: S23 S22 S21 S20
(A1A0) (B3B2) using 2-bit multiplier generates result: S13 S12 S11 S10
(A1A0) (B1B0) using 2-bit multiplier generates result: S03 S02 S01 S00

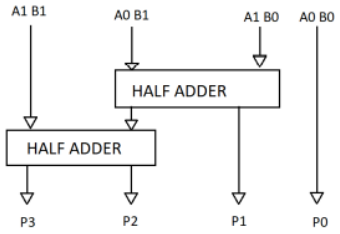


Fig 2.3 :12-bit binary multiplier

The 4-bit Quaternary Carry Look Ahead Adder (QCLA) is used to add three 4-bit data inputs: S23 S22 S21 S20 S13 S12 S11 S10 and S31 S30 S03 S02. The 4-bit Vedic multiplier is designed and the Fig. shows it. The last two MSBs of QCLA outputs are given as inputs to OR gate. In addition, the last stage 2-bit adder circuit through which the output value of OR gate can be controlled. One of the inputs to last stage 2-bit adder is obtaining from the output of or gate.

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

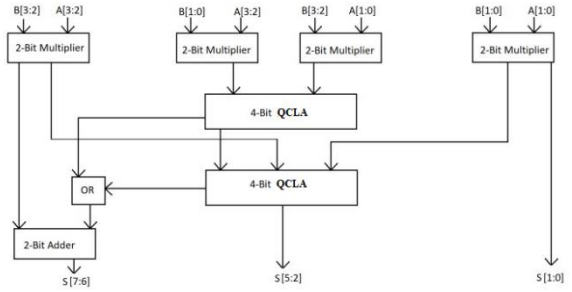


Fig 2.4 : 4-bit Vedic multiplier using QCLA

the 4-bit Vedic multiplier using QCLA is implemented and in the same way the size of the Vedic multiplier using QCLA is increased up to 32-bit i.e., 8, 16, and 32- bit.

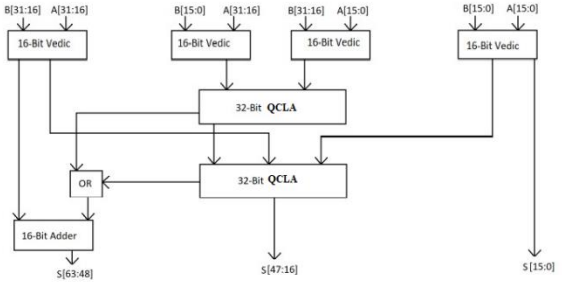


Fig 2.5: 32-bit Vedic multiplier using quaternary carry look ahead adder

2.2.1 VEDIC MEULTIPLIER USING QUATERNARY CARRY INCREMENT ADDER (PROPOSED DESIGN)

4.2.1 Carry Increment Adder

CIA_RCA The standard Carry Increment Adder (CIA) consists of RCA's and incremental circuitry. The incremental circuit is designed using HAs in ripple carry chain with a sequential order. The addition operation is done by separating the total number of bits in to group of 4bits and addition operation is performed by several 4-bit RCA's. Instead of computing two partial sums for each group and selecting the correct one, only one partial sum is calculated and incremented if necessary, according to the input carry. Thus, the second adder and the multiplexers in the carry-select scheme can be replaced by a much smaller incremental circuit and the modified architecture is the Carry Increment Adder (CIA). For example, an 8-bit CIA comprises of two 4-bit RCA. The first block of RCA adds first 4-bits to produce 4-bit partial sum and a carry output. Thus, first 4-bit of sum of CIA is directly obtained from first block of RCA. And the carry output of first RCA block is given as input to the cin of incremental circuit. Incremental circuit consists of Half Adders (HA). Hence, the partial sum obtained from the second RCA block is given to incremental circuit. The block diagram representation of an 8-bit CIA_RCA is as shown in Figure.

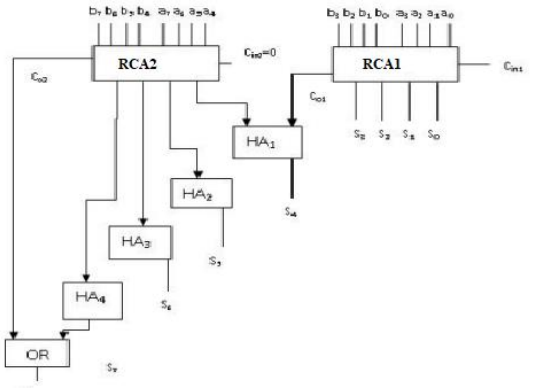


Fig 2.6 : Block diagram of CIA_RCA

We know that RCA is the basic binary adder circuit and is quite popular because of its simple design. However, it suffers from the worst propagation delay affecting the overall performance of the system. It is proved that CLA performs better than RCA in terms of delay at the expense of increased design complexity. We have modified CIA_RCA by replacing the RCA with CLA block. It is quite obvious because of the property of CLA, the overall delay performance will be improved. As similar to CIA_RCA incremental circuit can be designed using Half adders in ripple carry chain with a sequential order. The block diagram representation of CIA_CLA is as shown in Figure

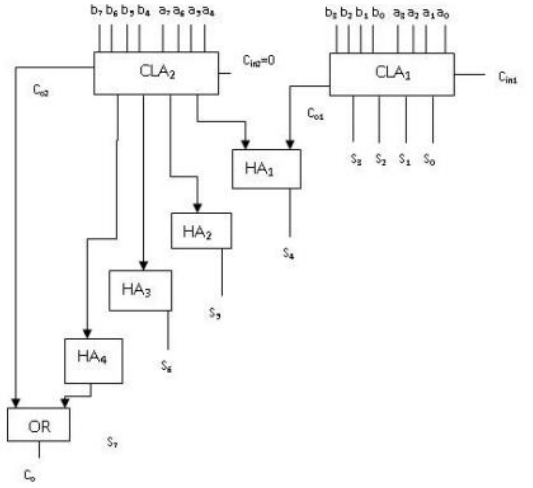


Fig 2.7: Block diagram of CIA_CLA

2.2.2 Vedic multiplier using Quaternary Carry Increment Adder (QCIA)

In this proposed design in the place of carry look ahead adder we use the quaternary carry look ahead adder, from the quaternary carry look ahead adder (QCLA) we can design the quaternary carry increment adder (QCIA). This quaternary carry increment adder makes the use in terms of area and delay due to these benefits we design the 32-bit Vedic multiplier by using the Quaternary Carry Increment Adder (QCIA)

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

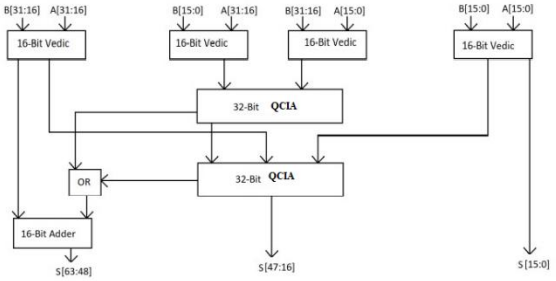


Fig 2.8: 32-bit Vedic multiplier using Quaternary Carry Increment Adder

RESULTS

New Project Summary:

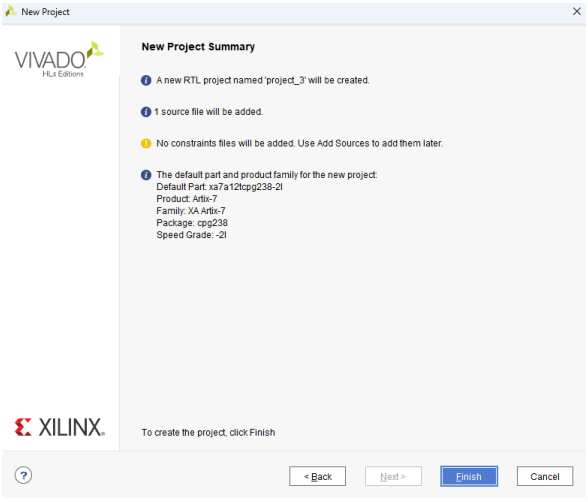


Fig 3.1. New project summary

Figure 3.1 shows the new project summary window displayed after creating a new RTL project in the Xilinx Vivado design suite.

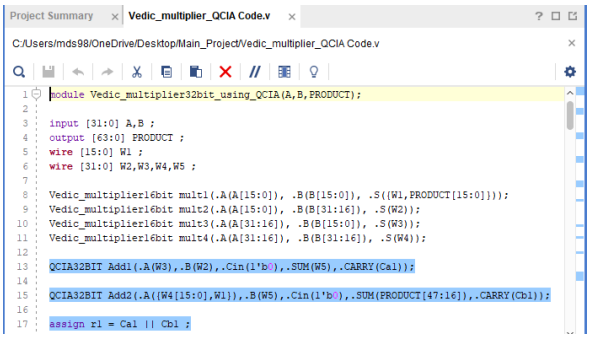


Figure 3.2 QCIA CODE

Figure 3.2 shows the Verilog HDL source code of the proposed 32-bit Vedic Multiplier using QCIA in Xilinx Vivado. The design consists of four 16-bit Vedic multiplier modules and two 32-bit QCIA adders, which are interconnected to generate the final 64-bit product. This code represents the RTL implementation of the proposed multiplier architecture before simulation and synthesis.

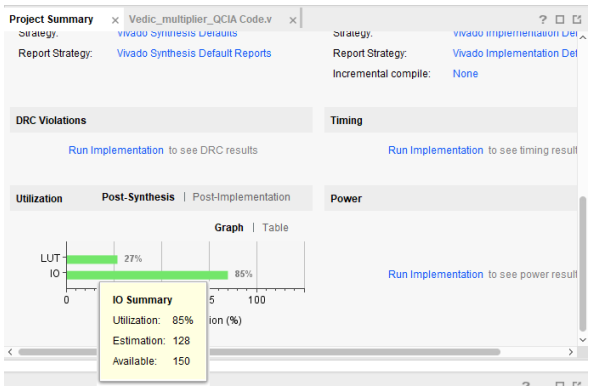


Figure 3.3 Post-Synthesis Summary

Figure 3.3 shows the post-synthesis resource utilization report generated by Xilinx Vivado for the proposed 32-bit Vedic Multiplier using QCIA. The report summarizes the estimated FPGA resource consumption, including LUT and I/O utilization, and provides access to timing, power, and DRC analysis after implementation. It is used to verify that the design efficiently utilizes the available FPGA resources.

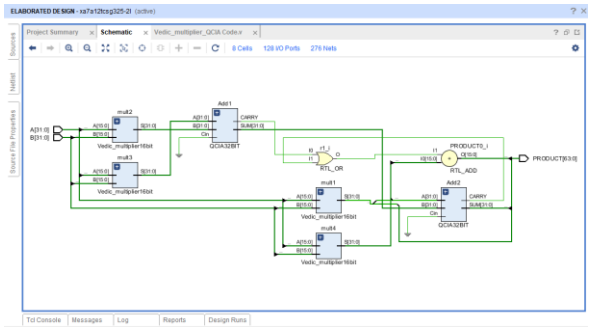


Figure 3.4. RTL Schematic of 32-bit Vedic Multiplier using QCIA

Figure 3.4 shows the RTL (Register Transfer Level) schematic of the proposed 32-bit Vedic Multiplier using the QCIA architecture generated in the Xilinx Vivado Design Suite. The schematic illustrates the hierarchical implementation of the design, where four 16-bit Vedic multiplier modules are interconnected with 32-bit QCIA adders and logic gates to produce the final 64-bit multiplication output. It provides a graphical representation of the data flow and module connectivity, confirming the correct integration of all submodules in the proposed multiplier architecture before synthesis and implementation.

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

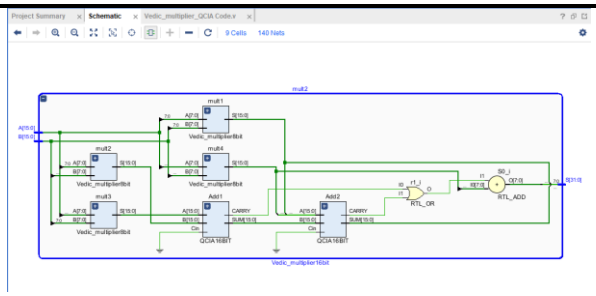


Figure 3.5. Elaborated RTL Design

Figure 3.5 shows the RTL (Register Transfer Level) schematic of the proposed 16-bit Vedic Multiplier using the QCIA architecture generated in the Xilinx Vivado Design Suite. The design is hierarchically implemented using four 8-bit Vedic multiplier modules**,** two 16-bit QCIA adders, and logic gates to combine the intermediate partial products and generate the final 32-bit multiplication output. The RTL schematic illustrates the interconnection of all functional blocks, verifying the correctness of the design architecture before synthesis and implementation.

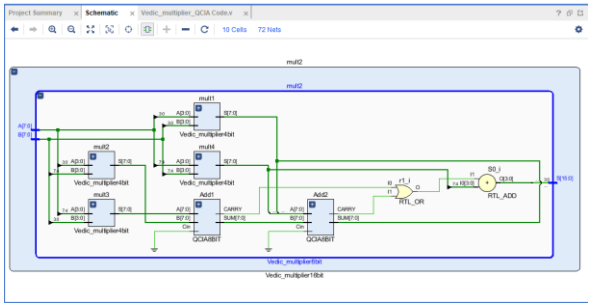


Figure 3.6. 8-bit RTL Schematic

Figure 3.6. shows the RTL (Register Transfer Level) schematic of the proposed 8-bit Vedic Multiplier using the QCIA architecture generated in the Xilinx Vivado Design Suite. The design is hierarchically constructed using four 4-bit Vedic multiplier modules**,** two 8-bit QCIA adders, and logic gates to combine the partial products and produce the final 16-bit multiplication output**.** The RTL schematic illustrates the interconnection of the functional blocks and verifies the correctness of the proposed architecture before synthesis and FPGA implementation.

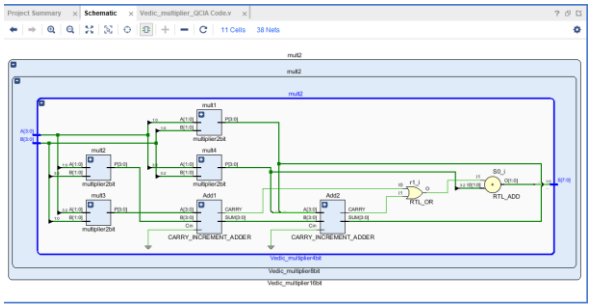


Figure 3.7. 4-bit RTL Schematic

Figure 3.7 shows the RTL (Register Transfer Level) schematic of the proposed 4-bit Vedic Multiplier using the QCIA architecture generated in the Xilinx Vivado Design Suite. The design is hierarchically implemented using four 2-bit multiplier modules**,** two 4-bit QCIA (Carry Increment Adder) blocks, and logic gates to combine the partial products and generate the final 8-bit multiplication output. The RTL schematic verifies the correct interconnection of all modules and confirms the functionality of the proposed design before synthesis and FPGA implementation.

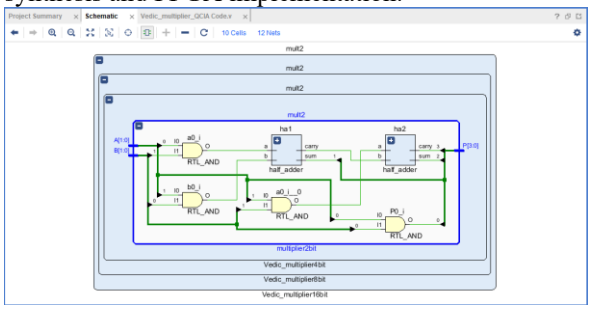


Figure 3.8. 2-bit RTL Schematic

Figure 3.8 shows the RTL (Register Transfer Level) schematic of the proposed 2-bit Vedic Multiplier generated in the Xilinx Vivado Design Suite. The design is implemented using AND gates and half adders to generate and combine the partial products, producing the final 4-bit multiplication output. The RTL schematic illustrates the data flow between the logic gates and adders, confirming the correct functionality of the basic multiplier module used as the building block for higher-bit Vedic multiplier architectures.

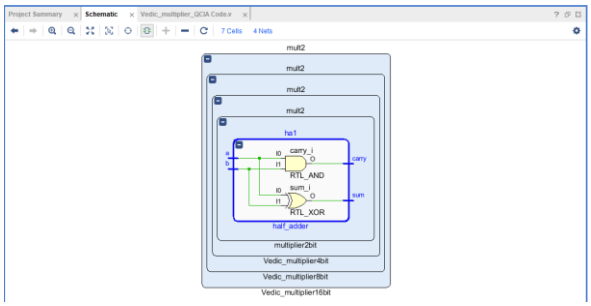


Figure 3.9. Half Adder RTL Design

Figure 3.9 shows the RTL (Register Transfer Level) schematic of the Half Adder generated in the Xilinx Vivado Design Suite. The circuit is implemented using one XOR gate to generate the Sum output and one AND gate to generate the Carry output. This basic arithmetic module serves as a fundamental building block in the proposed 2-bit Vedic Multiplier, contributing to the addition of partial products and ensuring correct multiplication operation.

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

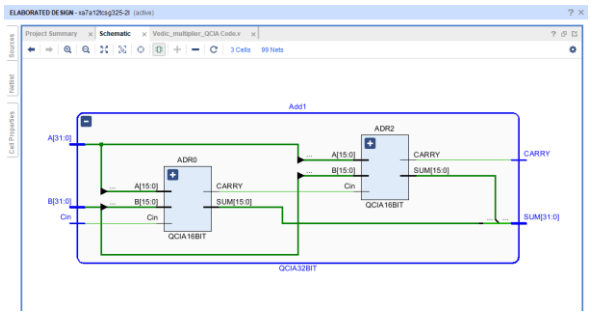


Figure 3.10. RTL Design of 32-bit QCIA

Figure 3.10 shows the RTL (Register Transfer Level) schematic of the proposed 32-bit Quaternary Carry Increment Adder (QCIA) generated in the Xilinx Vivado Design Suite. The design is hierarchically implemented using two 16-bit QCIA modules connected in cascade to perform 32-bit addition. The schematic illustrates the flow of input operands, carry propagation, and generation of the final 32-bit sum and carry output. This module serves as the high-speed adder in the proposed Vedic multiplier architecture, enabling efficient addition of partial products.

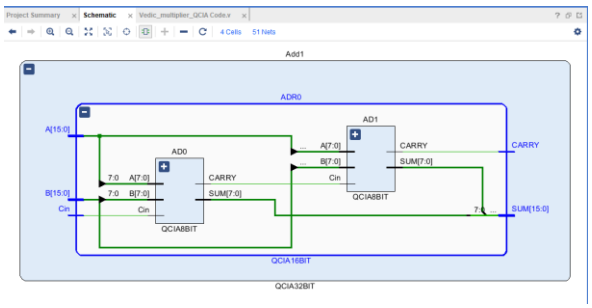


Figure 3.11. RTL Schematic of 16-bit QCIA

Figure 4.11 shows the RTL (Register Transfer Level) schematic of the proposed 16-bit Quaternary Carry Increment Adder (QCIA) generated in the Xilinx Vivado Design Suite. The design is hierarchically implemented using two 8-bit QCIA modules connected in cascade to perform 16-bit addition. The schematic illustrates the input operands, carry propagation, and the generation of the final 16-bit sum and carry output. This module is used as an intermediate building block in the proposed Vedic multiplier architecture to achieve faster and more efficient addition of partial products.

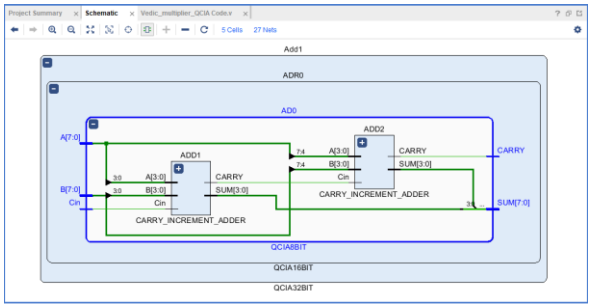


Figure 3.12. RTL Design of 8-bit QCIA

Figure 3.12 shows the RTL (Register Transfer Level) schematic of the proposed 8-bit Quaternary Carry Increment Adder (QCIA) generated in the Xilinx Vivado Design Suite. The design is hierarchically implemented using two 4-bit Carry Increment Adder (CIA) modules connected in cascade to perform 8-bit addition. The schematic illustrates the input operands, carry propagation, and the generation of the final 8-bit sum and carry output. This module is used as a building block in the higher-bit QCIA architecture to achieve faster addition in the proposed Vedic multiplier.

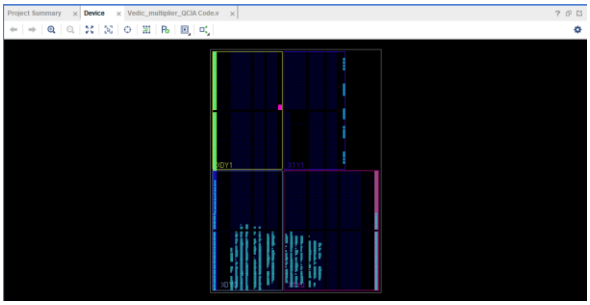


Figure 3.13. FPGA Device Utilization Layout

Figure 3.13 shows the FPGA Device Utilization Layout generated in the Xilinx Vivado Design Suite after implementing the proposed 32-bit Vedic Multiplier using QCIA.

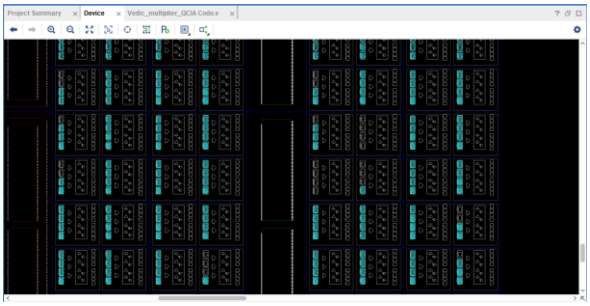


Figure 3.14. Detailed FPGA Device Placement

Figure 3.14 shows the detailed FPGA device placement of the proposed 32-bit Vedic Multiplier using QCIA in the Xilinx Vivado Design Suite.

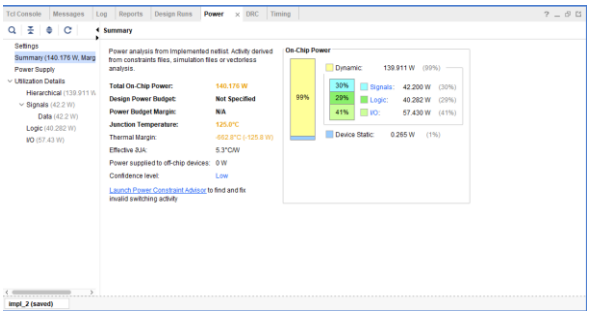


Figure 3.15. Power Analysis Report

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

Figure 3.15 shows the Power Analysis Report of the proposed 32-bit Vedic Multiplier using QCIA generated in the Xilinx Vivado Design Suite after implementation.

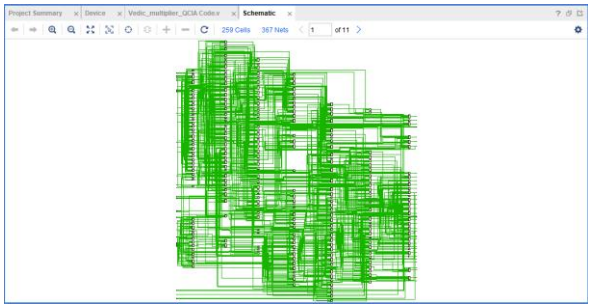


Figure 3.16. Gate-Level Schematic of 32-bit Vedic Multiplier using QCIA

Figure 3.16 shows the gate-level schematic of the proposed 32-bit Vedic Multiplier using the QCIA architecture generated in the Xilinx Vivado Design Suite after synthesis. The schematic represents the complete implementation of the design using interconnected logic gates and synthesized components. It illustrates the internal connectivity and data flow between the functional blocks, confirming the successful synthesis of the proposed architecture. This gate-level view is useful for verifying the structural implementation and overall hardware realization of the multiplier on the target FPGA.

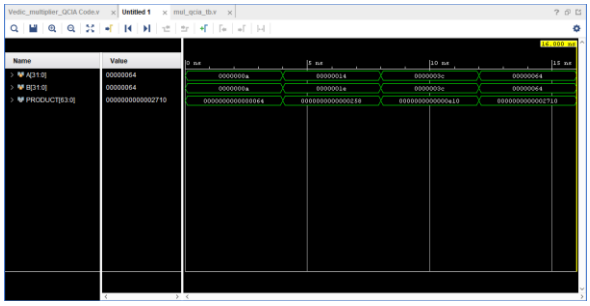


Figure 3.17. Simulation Waveform of 32-bit Vedic Multiplier using QCIA

Figure 3.17 shows the simulation waveform of the proposed 32-bit Vedic Multiplier using the QCIA architecture generated in the Xilinx Vivado Simulator.

4.1 RTL SCHEMATIC

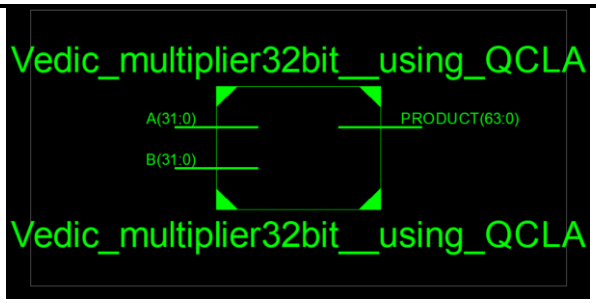


Fig4.1.1: RTL Schematic of existed Vedic multiplier using quaternary CLA

Fig 4.1.2: RTL Schematic of Proposed Vedic multiplier using quaternary CIA

4.2 TECHNOLOGY SCHEMATIC

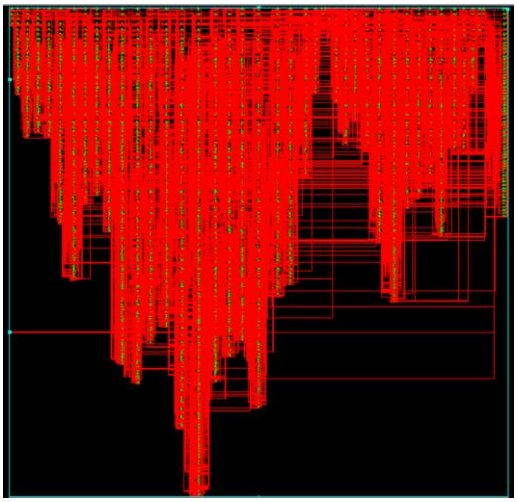


Fig 4.2.1: View Technology Schematic of Vedic multiplier using quaternary CLA (existed)

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

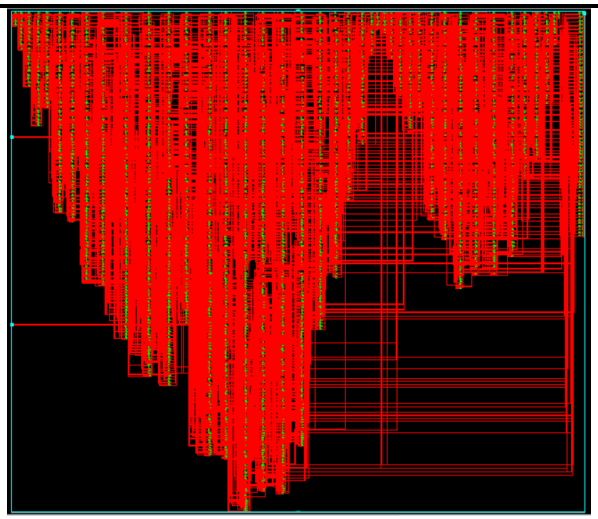


Fig 4.2.2: View Technology Schematic of Vedic multiplier using quaternary CIA (proposed)

4.3 SIMULATION

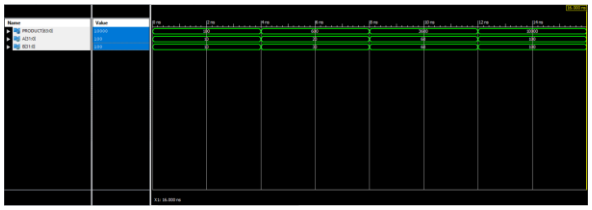


Fig 4.3.1: Simulated Waveforms of Vedic multiplier using quaternary CLA (existed)

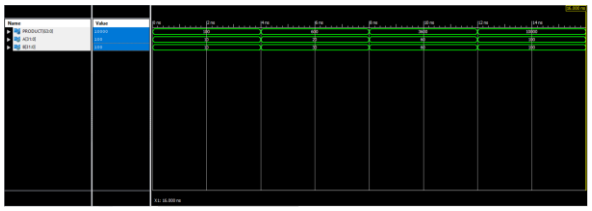


Fig 4.3.2: Simulated Waveforms of Vedic multiplier using quaternary CIA (proposed)

4.4 PARAMETERS

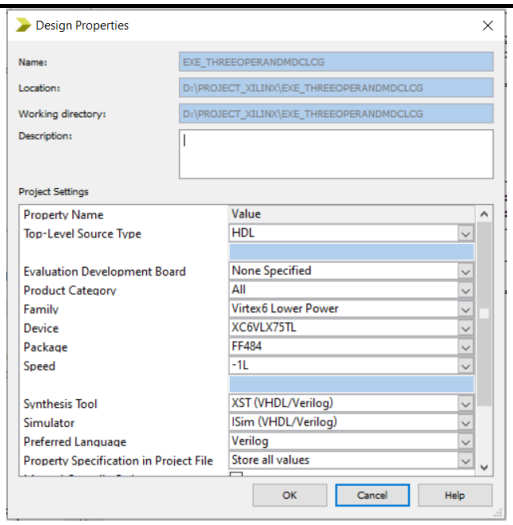


Fig 4.4.1: device used for synthesis

Parameter	Vedic multiplier using quaternary CLA (Existed)	Vedic multiplier using quaternary CIA (Proposed)
No of LUTs	2,003	1,949

Table: parameter comparison

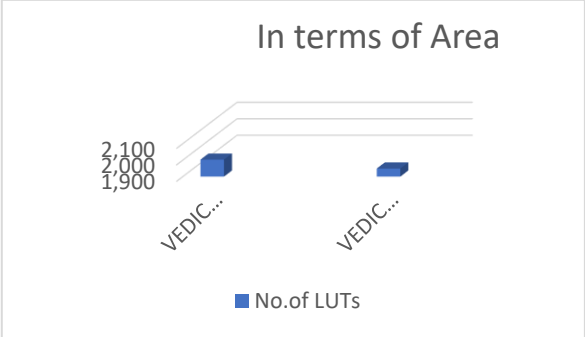


Fig 4.4.2: LUT comparison bar graph

CONCLUSION

Carry Increment Adder based on QSD is used in the design of Vedic multiplier in this proposed work. It is an area efficient and low power device. Carry propagation chain is eliminated by QSD number system. Adder speed is enhanced and system's computational time reduced by this design. In multiplier, circuit performance is enhanced using Vedic multiplier with CIA adder based on QSD. Xilinx-ISE 14.7 tool is used for simulating the proposed multiplier design. In this work from the above table, we can observe that number of LUTs required for existed design of Vedic multiplier using Quaternary Carry Look Ahead Adder (QCLA) is 2,003 and for proposed design of Vedic multiplier using Quaternary Carry Increment Adder (QCIA) is 1949. The analysis result shows that area required for QCIA multipliers is

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

low when compared with QCLA multiplier. Therefore, the proposed Vedic multiplier using Quaternary Carry Increment Adder (QCIA) is being used for area efficient devices

REFERENCES

[1] S. Mukherjee, D.K. Soni, Energy efficient multiplier for highspeed DSP application, Int. J. Comput. Sci. Mob. Comput. 4 (6) (June 2015) 66–75 pg.

[2] K. Mathew, S.A. Latha, T. Ravi, E. Logashanmugam, Design and analysis of an array multiplier using an area efficient full adder cell in 32 nm CMOS technology, Int. J. Eng. Sci. 2 (3) (2013) 8–16. [3] J. Ramasamy, S. Nagarajan, in: Hybrid Segment Approximate Multiplication for Image Processing Applications, Circuits and Systems, 2016, pp. 1701–1708.

[4] S. Srikanth, I.T. Banu, G.V. Priya, G. Usha, Low power array multiplier using modified full adder, in: 2016 IEEE International Conference on Engineering and Technology (ICETECH), IEEE, 2016, March, pp. 1041–1044.

[5] M.P.H. Langade, S.B. Patil, Design of improved systolic array multiplier and its implementation on FPGA, Ter-Natl. J. Eng. Res. Gen. Sci. 3 (6) (2015) 838–845.

[6] P.L. Thangkhiew, R. Gharpinde, P.V. Chowdhary, K. Datta, I. Sengupta, Area efficient implementation of ripple carry adder using memristor crossbar arrays, in: 2016 11th International Design & Test Symposium (IDT), IEEE, 2016, December, pp. 142–147.

[7] J. Barman, V. Kumar, Approximate carry look ahead adder (CLA) for error tolerant applications, in: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), IEEE, 2018, May, pp. 730–734.

[8] R.A. Javali, R.J. Nayak, A.M. Mhetar, M.C. Lakkannavar, Design of high-speed carry save adder using carry lookahead adder, in: International Conference on Circuits, Communication, Control and Computing, IEEE, 2014, November, pp. 33–36.

[9] P. Devi, A. Girdher, B. Singh, Improved carry select adder with reduced area and low power consumption, Int. J. Comput. Appl. 3 (4) (June 2010) 14–18.

[10] J. Ramasamy, N.S. Kumar, B. Sivasankari, Approximate carry select adder based multiplierless multiplication (AMLM), Middle-East J. Sci. Res. 24 (9) (2016) 2772–2777.

[11] J. Kaur, L. Sood, Comparison between various types of adder topologies, IJCST 6 (1) (2015) 62–66.

[12] P. Paliwal, J.B. Sharma, V. Nath, Comparative study of FFA architectures using different multiplier and adder topologies, Microsyst. Technol. (2019) 1–8.

[13] H.G. Tamar, A.G. Tamar, K. Hadidi, A. Khoei, P. Hoseini, High speed area reduced 64-bit static hybrid carry-lookahead/carry-select adder, in: 2011 18th IEEE International Conference on Electronics, Circuits, and Systems, IEEE, 2011, December, pp. 460–463.

[14] P. Balasubramanian, D.A. Edwards, H.R. Arabnia, Robust asynchronous carry lookahead adders, in: Proceedings of the International Conference on Computer Design (CDES), 2011, p. 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (World-Comp).

[15] P. Kishore, P.V. Sridevi, K. Babulu, K.P. Chandra, A novel low power and area efficient carry-lookahead adder using Mod-GDI technique, Int. J. Sci. Res. 4 (5) (2015) 1205–1210.

[16] S. Dubey, R. Rani, S. Kumari, N. Sharma, VLSI implementation of fast addition using quaternary signed digit number system, in: 2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), IEEE, 2013, March, pp. 654–659.

[17] A.N. Nagamani, S. Nishchai, Quaternary high performance arithmetic logic unit design, in: 2011 14th Euromicro Conference on Digital System Design, IEEE, 2011, August, pp. 148–153.

[18] A.N. Bankar, S. Hajare, Design of arithmetic circuit using quaternary signed digit number system, in: 2014 International Conference on Communication and Signal Processing, IEEE, 2014, April, pp. 793–797.

[19] D.P. Borkute, P. Patel, P.K. Dakhole, Delay performance and implementation of quaternary logic circuits, in: 2015 International Conference on Computing Communication Control and Automation, IEEE, 2015, February, pp. 1008–1012

[20] M.Sutha, Dr.R. Nirmala, Dr.PT. Kalaivani, Design Of Low Power Less Leakage Quarternary Adder Using Finfet, © 2022 IJCRT | Volume 10, Issue 5 May 2022 | ISSN: 2320-2882

[21] M.Bala Murugesh, S.Nagaraj, J.Jayasree, Modified High Speed 32-bit Vedic Multiplier Design and Implementation, Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020) IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4