VLSI Implementation of Adder Circuits Using Quaternary Signed Digit

Project Report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

By

SHUBHAM GUPTA (131079)

AKANKSHA MAHESHWARI (131102)

Under the Supervision of

Dr. SHRUTI JAIN

May-2017

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT, SOLAN - 173234, INDIA

TABLE OF CONTENTS

ACKNOWLEDGEMENT 5

LIST OF ACRONYMS & ABBREVIATIONS 7

LIST OF FIGURES 8

ABSTRACT 11

OBJECTIVE OF THE THESIS WORK 12

CHAPTER 1: INTRODUCTION 13

1.1 BASE-2 SIGNED DIGIT (BSD) 13

1.2 QUATERNARY SIGNED DIGIT (QSD) 13

1.3 FIELD PROGRAMMABLE GATE ARRAY (FPGA) 17

1.4 REQUISITE OF QSD 18

1.5 APPLICATIONS 18

CHAPTER 3: CONVERSIONS 22

2.1 DECIMAL TO QSD 22

2.1.1 METHOD 1- DIRECT METHOD (FOR POSITIVE CARDINAL DEPICTIONS) 22

2.1.2 METHOD 2- USING BASE-2 SIGNED DIGIT (2’S COMPLEMENT FORM FOR NEGATIVE CARDINAL DEPICTIONS) 23

2.2 QSD TO DECIMAL 24

2.2.1 DIRECT METHOD (FOR POSITIVE CARDINAL DEPICTIONS) 24

CHAPTER 4: ADDER AND SUBTRACTORS 25

4.1 BASIC ADDER UNIT 26

4.1.1 HALF ADDER 26

4.1.2 FULL ADDER 28

4.2 SIMULATION RESULTS 33

4.2.1 RESULTS FOR RIPPLE CARRY ADDER 34

4.2.2 RESULTS FOR CARRY SAVE ADDER 37

CHAPTER 5: PROPOSED WORK 41

5.1 CARRY FREE ADDITION 41

5.2 RULES FOR CARRY FREE ADDTION 41

5.3 EQUATIONS AND CIRCUMVOLUTION DIAGRAM 47

5.4 SIMULATION RESULTS 49

5.4.1 RESULTS FOR CARRY FREE ADDER 49

CHAPTER 6: CONCLUSION AND FUTURE WORK 53

6.1 CONCLUSION 53

6.2 FUTURE WORK 53

PUBLICATIONS: 56

DECLARATION BY THE SCHOLARS

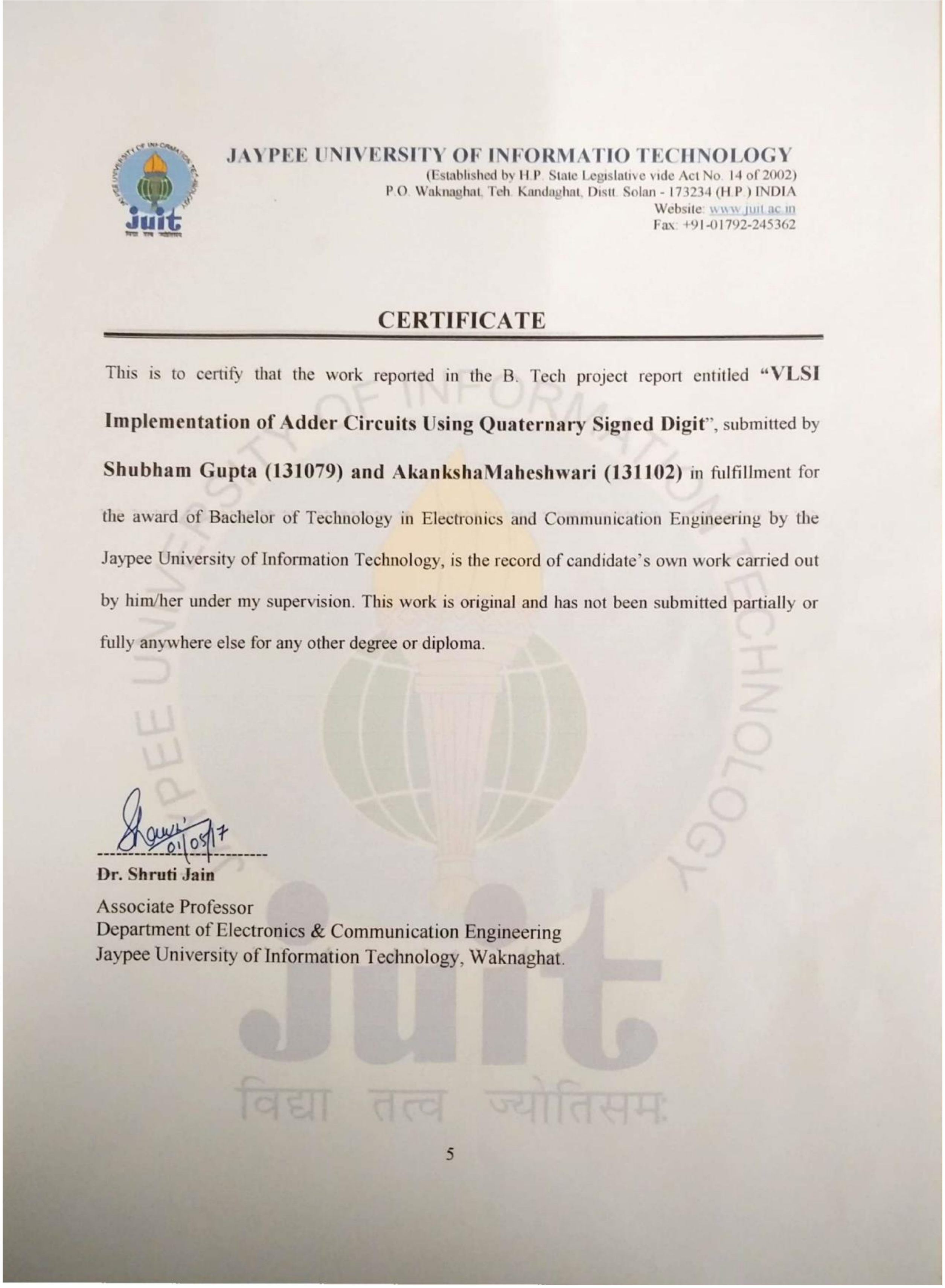
We hereby affirm that the work reported in the B-Tech thesis designated **“VLSI Implementation of Adder Circuits Using Quaternary Signed Digit”** submitted at **Jaypee University of Information Technology, Waknaghat, India,** is an authentic record of our work carried out under the supervision of **Dr. Shruti Jain**. It is ensured by us that the following work is not submitted elsewhere for any supplementary diploma or degree.

Shubham Gupta	Akanksha Maheshwari
(131079)	(131102)

Department of Electronics and Communication

Jaypee University of Information Technology, Waknaghat, India

Date: 1st May 2017



JAYPEE UNIVERSITY OF INFORMATIO TECHNOLOGY
(Established by H.P. State Legislative vide Act No. 14 of 2002)
P.O. Waknaghat, Teh. Kandaghat, Distt. Solan - 173234 (H.P.) INDIA
Website: www.juit.ac.in
Fax: +91-01792-245362

CERTIFICATE

This is to certify that the work reported in the B. Tech project report entitled **"VLSI Implementation of Adder Circuits Using Quaternary Signed Digit**", submitted by **Shubham Gupta (131079) and AkankshaMaheshwari (131102)** in fulfillment for the award of Bachelor of Technology in Electronics and Communication Engineering by the Jaypee University of Information Technology, is the record of candidate's own work carried out by him/her under my supervision. This work is original and has not been submitted partially or fully anywhere else for any other degree or diploma.

Shruti 01/05/17

Dr. Shruti Jain
Associate Professor
Department of Electronics & Communication Engineering
Jaypee University of Information Technology, Waknaghat.

ACKNOWLEDGEMENT

We have taken endeavors in this venture. Be that as it may, it would not have been conceivable without the kind support and help of numerous people and associations. I might want to extend my genuine because of every one of them.

We are considerably obligated to Dr. Shruti Jain for her direction and firm supervision and additionally to provide essential illuminating in regards to the venture and likewise for her support in finishing the venture.

We might want to express our appreciation towards our folks and individuals from Jaypee University of Information Technology for their kind co-engagement and consolation which help us in consummation of this venture.

We might want to express our uncommon appreciation and because of our HOD Prof. Sunil Vidya Bhooshan for giving us the chance of doing such a venture which gave us plentiful measure of learning background and information.

Our thanks and thanks additionally goes to our partners in developing the venture and individuals who have readily bailed us out with their capacities.

At long last we might want to acknowledge both co-creators of the venture for buckling down and making this venture genuine.

LIST OF ACRONYMS & ABBREVIATIONS

ALU	Reckoning Logic Unit
BCD	Base-2-Coded Decimal
BSD	Base-2 Signed Digit
CSA	Carry Save Adder
FA	Full Adder
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
HA	Half Adder
IIR	Infinite Impulse Response
K-map	Karnaugh-Map
LSB	Least Significant Bit
MOS	Metal-Oxide Semiconductor
MSB	Remarkably Significant Bit
MVL	Multi Valued Logic
OPAMP	Operational Amplifier
QSD	Quaternary Signed Digit
RAM	Random Access Memory
RBSD	Ripple Carry Base-2 Signed Adders
RCA	Ripple Carry Adder
SEED	Self Electro-optic Effect Devices
VHDL	Verilog Hardware Description Language
VLSI	Very Broad Scale Integration

LIST OF FIGURES

Figure 1 Half adder logic diagram 27

Figure 2 Block Diagram for Half Adder... 27

Figure 3 Block diagram of quaternary half adder circumvolution.. 28

Figure 4 Full adder circumvolution diagram.. 30

Figure 5 Ripple Carry Adder Block Diagram.. 31

Figure 6 Carry Save Adder Circumvolution.. 33

Figure 7 Simulation Result of Ripple Carry Adder.. 34

Figure 8 Simulation Result of Ripple Carry Subtractor 34

Figure 9 Synthesis Report Ripple Carry Adder.. 35

Figure 10 Time Lag for Ripple Carry Adder.. 36

Figure 11 Simulation Result of Carry Save Adder 37

Figure 12 Synthesis Report Carry Save Adder 38

Figure 13 Time Lag for Carry Save Adder 39

Figure 14 Block Diagram of Carry Free QSD Addition 41

Figure 15 Gates combinational circuit diagram for single digit QSD adder cell.................. 48

Figure 16 Simulation Result of QSD Adder 49

Figure 17 Synthesis Report of Carry Free Adder.. 50

Figure 18 Time lag for Carry Free Addition.. 51

Figure 19 FPGA version details... 52

LIST OF TABLES

Table 1 Converting 182 into QSD.. 22

Table 2 Truth Table for Half-Adder... 27

Table 3 Truth Table for Full Adder.. 30

Table 4 Comparison Table for Ripple Carry Adder.. 37

Table 5 Comparison Table for Carry Save Adder.. 40

Table 6 Intervening sum and carry from -6 to +6... 42

Table 7 Alteration between the inputs and final turnouts of the Intervening carry and Intervening sum.. 43

Table 8 Addition of two QSD cardinal depictions.. 46

Table 9 Comparison Table for Carry Free Adder... 51

ABSTRACT

The data manipulating agility is hampered in base 2 cardinal depiction methodology because of reproduction and evolution of trajectory of peculiarly when the cardinal depiction of bits upturns.

In this project we have successfully implemented and observed that the carry is generated in the Ripple Carry Adder (RCA) and the Carry Save Adder (CSA) and thus we aim to remove the generated trajectory which we have contended in this work. We face O(n) trajectory breeding lagging n-bit base-2 process. To resolve this complication, signed cipher notation is utilized for trajectory complimentary reckoning engagements. Subtraction without borrow as well as aggregation without trajectory can be implemented using Quaternary Signed Digit (QSD) cardinal depiction methodology. In this work we extend the QSD addition to eliminate trajectory from aggregation and lagged aggregation and we will also work on minimizing the power consumption and lag at a particular frequency. We will also be comparing the reckoning engagements of QSD cardinal depiction methodology with that of base-2 cardinal depiction methodology. For all these delineating and calculations we will be using Xilinx 14.7 software.

OBJECTIVE OF THE THESIS WORK

In this project we have mainly focused our work on the Quaternary Signed Digit Methodology. We have worked on how to represent a decimal cardinal depiction into its respective QSD cardinal depiction and vice versa using different conversion techniques. The main objective of doing this project is aggregation of two QSD cardinal depictions which may be both positive, both negative or one positive and one negative respectively. We have shown aggregation of two QSD cardinal depictions using different examples to clarify the technique availed. In future, we will extend our work from aggregation to subtraction as well as augmentation of two QSD cardinal depictions.

CHAPTER 1: INTRODUCTION

Signed-cipher notation illustration won’t be able to accomplish quick aggregation of numerals as a result of it will get rid of the strings of reliant carries. Within the base-2 cipher scheme, an unusual scenario of signed-cipher notation illustration is non adjoining kind, which might supply rapid edges with marginal area.

1.1 BASE-2 SIGNED DIGIT (BSD)

A base-2 marked figure documentation depiction of a whole number $k \in [0,2n-1]$ is a base-2 depiction symbolized by (Kn, Kn-1, . . ., K0) BSD where $K_i \in \{-1, 0, 1\}$.We will call the K is marked bits, or s bits for short and - 1 will be composed as 1. A whole number can have impressive BSD depiction. For instance, k = (8)10 can be drafted as (01000) in BSD. In the midst of the plausible BSD depiction of a whole number there are two elite depictions. Introduce day centralized computers are arranged on base-2 factual plan. It has 2 conceivable states "0" and '1'. In such plan, "1" + "1" is "0" with direction "1" (i.e. 1+1=10). The direction produced ought to then be included with one more "1", as an outcome a supplementary direction "1" is created which sets up the slack intricacy in PC circumvolutions.

1.2 QUATERNARY SIGNED DIGIT (QSD)

Engagements involving aggregation, subtraction and augmentation form the basic reckoning engagements which are broadly availed and execute a crucial aspect in discrete cipher notational electronic devices such as pcs and signal processors. Formidable reckoning is very compelling as the aggregators availed in the methodology concludes the agility of the cipher notational processor. This also gratifies as an architecture for the amalgamation of all the various reckoning engagements. In Cipher notational Signal Processing, adder is availed to execute FIR, IIR algorithms. Lag in breeding time, finite bits and formidable intricacy of the circumvolution are the extensive dilemma’s faced in reckoning engagements. Trajectory chains which are propagated in the adder are responsible for time lag in the circumvolution. Some of the circumspections of BSD cardinal depiction methodologies are agile due to taking into account which restricts breeding and forming of trajectory specifically when the cardinal depiction of bits gets boost up. Hence, low cache density and broad intricacy is experienced. Reckoning without generating trajectory can be done using Quaternary Signed Digit (QSD) which is a formidable radix cardinal depiction methodology. Present research shows that, breeding chain is eradicated in QSD cardinal depiction methodology cavilled by trajectory, emerging in devaluation of taking into account time productively, bettering the pace of the methodology. Quaternary Signed Digit (QSD), a formidable radix based signed cipher notation cardinal depiction methodology, allows formidable collective data storage density and barely intricacy [2]. Signed cipher notation portrayal is the one which we can avail to attain agile aggregation of numerals as a result of it can dispose of carries.[2]Portrayal of a signed-cipher notation decimal cardinal depiction D in conditions of an ‘n’ cipher notation QSD as:

$$D = \sum_{i=0}^{n-1} X_i 4^i \qquad (1)$$

The agility of the artificial intelligence of processors banks on remarkably the agility of the aggregators availed in the methodology. Remarkably of the reckoning procedures endures from dilemmas like finite cardinal depiction of bits, circumvolution’s intricacy and lag. Convey look ahead viper deliver scarcely a reproducing slack, however it is limited to insufficient cardinal delineation of bits because of the circumvolution multifaceted design. We battle an impressive light-footed QSD aggregator which is having the capability of working direction complimentary total and get complimentary subtraction utilizing QSD cardinal delineations. For any operand estimate the QSD accumulation/subtraction engagement controls a built up cardinal portrayal of min terms. In QSD cardinal portrayal Methodology direction reproducing chains are prohibited or neglected which downsizes the considering time. QSD is shows additionally profits if there should be an occurrence of parallelism and door multifaceted design. Outlining this retribution unit utilizing QSD cardinal delineation depiction has connected with the engagement of a few researchers. Besides, unique advances in greetings tech specialized information for blended circumvolutions make expansive scale figuring circumvolutions are ample for VLSI achievement. In this venture, we fight a considerable readiness QSD snake which is versatile of direction complimentary collection, obtain complimentary subtraction. The QSD aggregation/subtraction engagement fascinates an established cardinal depiction of min terms for any operand size. In QSD cardinal depiction methodology trajectory breeding chain are excluded or disregarded which deflates the taking into account time mainly, thus exaggerating the agility of the machine. Signed cipher notation cardinal depiction methodology offers the contingency of trajectory complimentary aggregation. QSD Adder / QSD Multiplier circumvolutions are logic circumvolutions delineated to enforce formidable-agility reckoning engagements. In QSD cardinal depiction methodology trajectory breeding chain are excluded or disregarded which deflates the taking into account time mainly, thus exaggerating the agility of the machine. Quaternary Signed Digit (QSD) cardinal depictions whose radix is 4 are availed in reckoning engagements to effectuate the trajectory complimentary reckoning engagements. The extent of QSD cardinal depiction is from -3 to 3. In any n cipher notation QSD cardinal depiction, each cipher notation can be characterized by a cardinal depiction from the cipher notation set [-3,-2,-1, 0, 1, 2, 3]. Trajectory complimentary aggregation and supplementary reckoning engagements on broad cardinal depiction of cipher notations such as 64, 128, or more can be implemented with the established firm lag and barely intricacy. In base-2 cardinal depiction methodology trajectory is an extensive dilemma in reckoning engagement. With the base-2 cardinal depiction methodology, the taking into account agility is finite by forming and breeding of trajectory in particularly as the cardinal depiction of bits increases. A trajectory complimentary reckoning engagement can be effectuated using a formidable radix cardinal depiction methodology such as Quaternary Signed Digit (QSD). In QSD, each cipher notation can be characterized by a cardinal depiction from -3 to 3. Trajectory complimentary aggregation and supplementary engagements on a broad cardinal depiction of cipher notations such as 64, 128, or more can be implemented with firm lag and barely intricacy. Besides, writing surveys reasons that multi-esteemed rationale (MVL) would be a superior decision to approach the situation of developing dashing chips for working hustling considering engagement. Quaternary Signed Digit (QSD) have a broad commitment in impressive radix (=4) direction complimentary retribution engagement. For figure notational achievement, the marked figure documentation quaternary cardinal delineations are described utilizing 3-bit 2's compliment documentation. In this venture, a straightforward and new method of base (2's compliment) to QSD transformation is proposed and explained. Outline of base-2 rationale circumvolutions is effortlessly conceivable when the interlinked is scarcely. As the cardinal portrayal of sources of info expands the interlinked is repetitive work. The unpredictability increments, as chip territory is emptied. To topple the quandary multi-esteemed rationale is conceivable arrangement. The Quaternary Signed Digit satisfies us to expand the rationale levels. This cardinal portrayal philosophy based rationale circumvolutions gives scarcely slack as compared to the base-2 circumvolutions. In this paper the retribution unit is portrayed which gives preferred outcomes over the base-2 circumvolutions. The Quaternary Signed Digit based collection averts undulating of direction. This evacuates the total direction controlling circumvolution, which empowers to fabricate quick aggregators which can be benefited in supplementary processors to give quick outcomes. These aggregators give the firm slack invariant to the cardinal delineation of sources of info. Utilizing this snake, multipliers are outlined which gives hustling comes about and the plan are scarcely complicated. Figuring engagements experience from known difficulty's understanding limited cardinal portrayal of bits, reproducing time slack, and circumvolution multifaceted nature. Convey look ahead enhances the rearing slack, yet is limited to an insufficient cardinal portrayal of figure documentations because of the multifaceted nature of the circumvolution. Marked figure documentation cardinal delineation system offers the possibility of Trajectory complimentary accumulation. We fight an imposing dexterity QSD figuring rationale unit which is versatile of direction complimentary collection, get complimentary subtraction. The QSD aggregation/subtraction engagement fascinates an established cardinal depiction of min terms for any operand size. In QSD cardinal depiction methodology trajectory breeding chain are excluded or disregarded which deflates the taking into account time mainly, thus exaggerating the agility of the machine. For operating any engagement in QSD, first convert the base-2 or any supplementary input into quaternary signed cipher notation. A cardinal depiction can be characterized in two forms that is signed and unsigned cardinal depiction. A signed cardinal depiction consists of magnitude and sign where as an unsigned cardinal depiction has only magnitude but not sign.

1.3 FIELD PROGRAMMABLE GATE ARRAY (FPGA)

An FPGA is IC, a semiconductor device which can be reprogrammed by customer according to his requirement after manufacturing. It comprises of array of configurable logic blocks (CLB's) and switches. Logic designs can be implemented on FPGA's (Xillinx Spartan-6 XC6SLX45 FPGA) by using either VHDL or Verilog code on Xilinx Simulator [3].

Attentive rationale were benefited to manufacture techniques comprised of many chips that are associated with wires in the mid-1960s. It was exorbitant and tedious reshaping such philosophy's which required modifying the board. Chip fabricator presented Programmable Logic Device (PLD) i.e. a solitary chip and constituted of a variety of detached AND–OR entryways. The PLDs grasped a variety of breakers that could be blown open or left shut to interface various contributions to each AND door. Since PLDs could deal with up to 20 rationale conditions outlining complicated strategies utilizing numerous PLDs was a testing procedure. To handle this quandary, chip creators presented Intricate PLDs (CPLD) and FPGA.A CPLD constituted of group of PLD hinders whose sources of info and last turnouts are represented by worldwide interlinked lattice. CPLDs give two levels of re-configurability; reconfiguring the PLD squares and interlinked between them. The structure of FPGAs is not quite the same as that of CPLDs. The FPGAs are constituted of a variety of basic and Configurable Logic Blocks (CLBs) and switches that are used to decide the associations between CLBs. Keeping in mind the end goal to execute a calculation in the FPGAs, each CLB is orchestrated independently first and after that changes are organized to associate or separate CLBs. In spite of the fact that there are diverse strategies for associating and separating CLBs, the amazingly broadly profited procedure depends on utilization of RAM/glimmer switches. In this strategy, static RAM or blaze bits are profited to control the pass transistors for each interlinked. For example, the switch can be shut or opened by stacking bit 1 or 0, individually. Since the current FPGAs can contain up to 10 million doors, manual control of switches is unthinkable. In this manner, FPGAs fabricator give advancement programming's that take rationale configuration as information and afterward last turnout a bit stream, which arranges the switches.

FPGA gives good, formidable agility clock, formidable bandwidth, and facilitates simultaneously multiple engagements. FPGA's has its applications in various fields like speech recognition, computer hardware emulation, metal detection etc.

1.4 REQUISITE OF QSD

The essential for impressive deft figure notational circumvolutions turned out to be all the more remarkable as sensible mixed media and correspondence applications fusing advising handling and considering. The downside of current PCs prompt the disintegration in implementation of retribution engagements, for example, accumulation, subtraction, division, growth on the parts of direction reproducing time slack, impressive power utilization and wide circumvolution multifaceted nature. This venture investigates the direction complimentary n figure documentations conglomeration/subtraction as the direction reproducing slack is astoundingly essential variable with respect to the nimbleness of any figure notational technique.[11]

1.5 APPLICATIONS

Cipher notational methodology's play a outstanding role in day to day life. There are various applications in the field of cipher notational methodology such as computers, process controllers, signal processors, computer graphics, image processing, optical taking into account in the optoelectronic devices such as lasers, array illuminators, SEED (self-electro-optic effect devices), spatial light modulators in which formidable agile reckoning engagement became significant due to the spread of wire barely communication and manageable taking into account methodology.[12] Aggregators are astoundingly as often as possible benefited in different electronic applications e.g. Figure notational flag preparing in which aggregators are profited to uphold different calculations like FIR, IIR and so forth.

CHAPTER 2: LITERATURE REVIEW

Application	Author	Year	Work
carry free n digits addition/subtraction[1]	PRANALI S. KAMBLE(IJAEEE)	2012	• Redundant binary signed digit cipher notations had been utilized as one of the SD depiction for high agility VLSI multiplication. • Two bit natural depiction of binary logic is availed for each Quaternary cipher notation and aggregation is performed in binary itself. • Depicted using 3-bit 2's complement notation. • QSD negative cipher notation is the QSD complement of the QSD positive cipher notation.
carry free addition, borrow free subtraction.[2]	BHUKYA RAJESH(IJPRES)	August 2014	• To depict a numeric value N log4N cipher notation of QSD digits and 3 log4N binary bits are required while for the same $\log_2 N$ BSD digits and 2 log2N binary bits are required in BSD depiction. • Advantages of both parallelisms as well as reduced gate complexity. • To change over n-bit paired information to its comparable q-digit QSD information, we need to change over this n-bit twofold information into 3q-bit double information Using 6 variable K-map, the logic equations specifying a minimal hardware realization for generating the intervening carry and intervening sum are derived. • QSD adder using NAND-NAND implementation for single digit aggregation, the dynamic power dissipation is 36.255W & 5GHz frequency. • Delay of the proposed design is 2ns.

Divider made by using QSD technique for fast division without carry propagating delay.[3]	SANDEEP KAUR (IJESRT)	August 2015	• Implementation of large digits of digital such as 64, 128 or more cipher notations can be implemented with constant delay. • Steps of QSD division are first take two binary cipher notations for the division. One binary cipher notation is dividend and other is divisor. Then convert the dividend and divisor into QSD cipher notation system. After conversion, shift divisor right and compare it with current dividend. If divisor is larger, shift 0 as the next bit of the quotient. If divisor is smaller, subtract to get new dividend and shift 1 as the next bit of the quotient. • The power availed by QSD divider is 23.50Mw and total delay is 18.06ns.
carry free aggregation[4]	KAVITA KUNTAL	February 2014	• Quaternary is the base 4 redundant cipher notation system. The degree of redundancy usually increases with the increase of the radix. • QSD adder using NAND-NAND implementation for single digit aggregation, the dynamic power dissipation is 36.255W & 5GHz frequency. • Delay of the proposed design is 2ns.
binary (2's compliment) to QSD conversion is proposed and described[5]	TANAY CHATTOPADHYAY	December 2012	• Quaternary logic is based on radix-4 cipher notation system. • Multi Valued Logic is availed. • Design of a prototype of digital to analog converter circuit using operational amplifier (OPAMP) to verify the technique.

			• To convert n-bit binary data to its equivalent q-digit QSD data, we have to convert this n-bit binary data into 3q-bit binary data. • Proposed circuit can be designed with v MOS- OPAMP.
QSD aggregation using delayed aggregation technique[6]	S.MALLESH(IJEEE)	October 2014	• Delayed aggregation in place of carry free aggregation. • By using Wallace trees to accumulate results without carry propagation overhead. • Design of floating point adder in Verilog HDL in Xilinx ISE environment based on Spartan 3E FPGA family.
carry free aggregation and borrow free subtraction QSD Multiplier[7]	SHRIKESH A. DAKHANE(IJCSIT)	2014	• The partial product generator and single bit multiplier is the key elements of the design. • Multiplication is done with partial product generators and QSD adders. • Increasing the cipher notation of bits of operation it gives constant delay of 4.287ns for 4bit QSD adder. • For 4x4 QSD multiplier, delay is found out to be 11.258 ns.
high agility QSD adder is proposed which is capable of performing carry free aggregation and borrow free subtraction using QSD cipher notations[8]	JYOTI R HALLIKHED(IJEEE)	May 2015	• QSD cipher notation uses 25% less space than BSD to store cipher notation. • The 3q-bit binary data is converted from the n-bit binary data, thus equivalent q- digit QSD data is getting from conversion of n-bit binary data. • The QSD ALU design performance is better comparing to other designs. • The QSD adder complexity is linearly proportional to the cipher notation of bits which are of the same order as the simplest BCD and other adder, such as the RCA.

CHAPTER 3: CONVERSIONS

We typically deal in decimal cardinal depiction methodology in real lives as decimal cardinal depiction methodology is more practical and understandable in general. But here we are dealing in Quaternary Signed Digits we should be able to convert decimal cardinal depictions into QSD. So here we will be explain the procedure to covert various cardinal depiction methodology into QSD.

2.1 DECIMAL TO QSD

Decimal cardinal depiction methodology being the remarkably popular and simplest is easy to understand. So here we will apply our knowledge of QSD cardinal depiction methodology and discuss various methods to convert decimal cardinal depiction methodology into QSD cardinal depiction methodology. [13]

2.1.1 METHOD 1- DIRECT METHOD (FOR POSITIVE CARDINAL DEPICTIONS)

As we are aware of the conversion of decimal cardinal depiction into base-2 using division method. So we apply the same technique of long division on decimal cardinal depiction to convert it into its respective QSD portrayal.[14] An example will make it clearer. Let us consider the following example:

$(182)_{10}$.

Table 1 Converting 182 into QSD

4	182	2
4	45	1
4	11	3
	2	

Therefore $(182)_{10} = (2312)_{QSD}$

2.1.2 METHOD 2- USING BASE-2 SIGNED DIGIT (2'S COMPLEMENT FORM FOR NEGATIVE CARDINAL DEPICTIONS)

1 digit notation QSD cardinal depiction can be characterized using a 3-bit base-2 equivalent as

$\bar{3}$=101

$\bar{2}$=110

$\bar{1}$=111

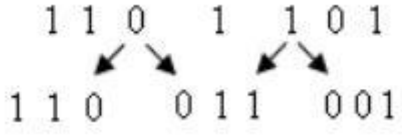
0 = 000

1 = 001

2 = 010

3 = 011

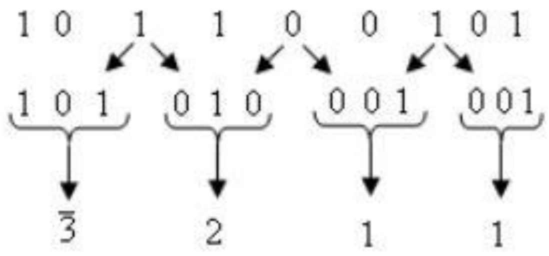
In this way, to change over n-bit base-2 information into its identical q-figure documentation QSD information, the n-bit base-2 information must be changed over into3q-bit base-2 information. To effectuate this we should part the third, fifth, seventh piece .i.e. odd piece (from LSB to MSB) into two sections. Be that as it may, we can't part the MSB bit. In the event that the odd piece is 0 then, it is part into 0 and 0 and on the off chance that it is 1 then, it is part into 1 and 0. [15]
An illustration makes it clear, the part procedure of a base-2 cardinal portrayal (1101101)2 is demonstrated as follows:



So we need to part the given base-2 information q-times. For instance, the part is one time for change of a 2-figure documentation quaternary cardinal delineation, the part is two times for transformation of a 3-figure documentation quaternary cardinal portrayal et cetera. In each part one additional piece is created. For change from base-2 to QSD, the required cardinal portrayal of base-2 bits are

$$n = 3q - \{1 \times (q-1)\} \qquad (2)$$

Thus, to convert base-2 cardinal delineation into its equal QSD the cardinal portrayal of bits ought to be 3,5,7,9 and so on. As per the above condition each 3-bit can be changed over into its QSD. Let $(-155)_{10} = (101100101)_2$ have be changed over to base-2 first (in 2's supplement shape and after that to its identical QSD. "$(101100101)_2$" is 9-bit base-2 information. Its third piece is 1, fifth piece is 0 and seventh piece is 1. So from the condition (2) we can state that, its QSD equal is of4-figure documentation. Consequently as per the part strategy expressed over the base-2 information can be communicated as take after [16]



2.2 QSD TO DECIMAL

Now we will be discussing the reverse of what we discussed in the antecedent section. Here we will discuss how to convert QSD to decimal by taking various examples to build our understanding better.

2.2.1 DIRECT METHOD (FOR POSITIVE CARDINAL DEPICTIONS)

As we have discussed above the formula using equation (1) for conversion of QSD to decimal which is:-

Let us consider the example:-

$(301)_{QSD}$, now to convert this to decimal cardinal depiction we consider the above formula

$$(301)_{QSD} = 3\times 4^{2} + 0\times 4^{1} + 1\times 4^{0}$$

$$= 48 + 0 + 1$$
$$= (49)_{10}$$

CHAPTER 4: ADDER AND SUBTRACTORS

An adder–subtractor is a circumvolution that is versatile of including or subtracting cardinal delineations (specifically, base-2). The following is a circumvolution that does including or subtracting depending a control flag. It is additionally conceivable to build a circumvolution that upholds both collection and subtraction in the meantime. [17]

4.1 BASIC ADDER UNIT

The amazingly essential retribution engagement is the total of 2 base-2 figure documentations, i.e. Bits. A combinable circumvolution that includes 2 bits, agreeing the hypothesis masterminded beneath, is known as a half viper (HA). A full snake is one that includes 3 bits, the third is created from the predecessor total engagement i.e. from the HA.[18] An approach for executing a full viper is to use two half aggregators in its achievement. There are two sorts of essential viper units as talked about beneath:

4.1.1 HALF ADDER

A half adder is utilized to include two base-2 figure documentations *A* and *B* inside and out. It yields *S*, the accumulation of *A* and *B*, and in parallel likewise yields direction C_0. In spite of the reality, a half viper is not hugely invaluable, it can be passed-down as a building hinder for bigger including circumvolutions, for example, full aggregators (FA).[19] One possible approach to yield a last turnout is utilizing two AND doors, two inverters, and an OR entryway rather than a XOR door as appeared in Fig. 1. Boolean Equations:

$$S = AB = \overline{A} \cdot B + A \cdot \overline{B} \qquad (2)$$

$$C_0 = A \cdot B \qquad (3)$$

To understand what a half adder is it urges the requisite to learn about the aggregators. Adder circumvolution is a combinable cipher notational circumvolution that is availed for adding different numerals. A model snake circumvolution yields a total piece (meant by S) and a direction bit (signified by C) as the last turnout. As a rule aggregators are acknowledged for including base-

2 cardinal delineations however they can be likewise acknowledged for including elective arrangements like BCD (base-2 coded decimal), XS-3, and QSD and so forth. Other than accumulation, snake circumvolutions can likewise be used for a great deal of supplementary applications in figure notational hardware like as approach deciphering, table file estimation and so on. Snake circumvolutions are of two sorts: Half viper advertisement Full viper. Full aggregators are illustrated inside the following area, in this segment we will concentrate on half aggregators specifically.[20]

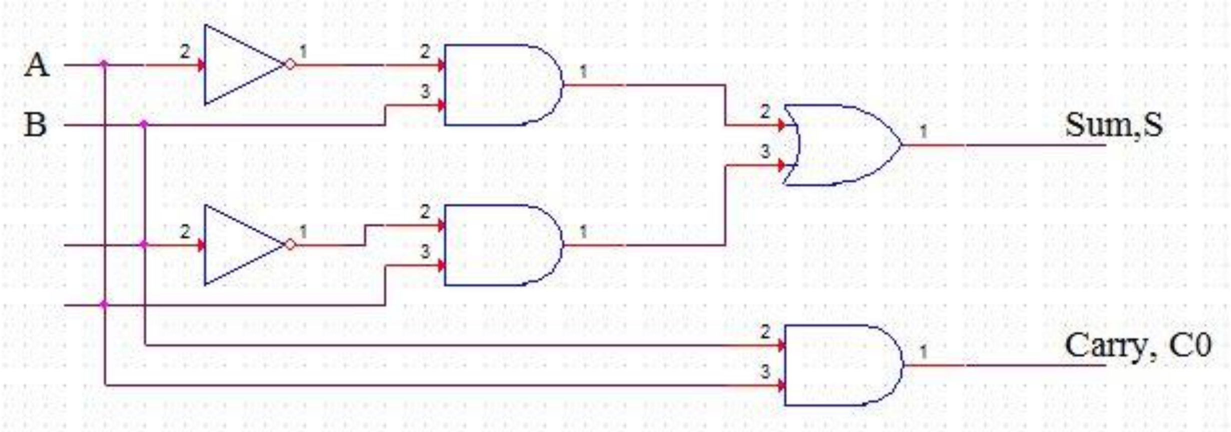


Figure 1 Half adder logic diagram

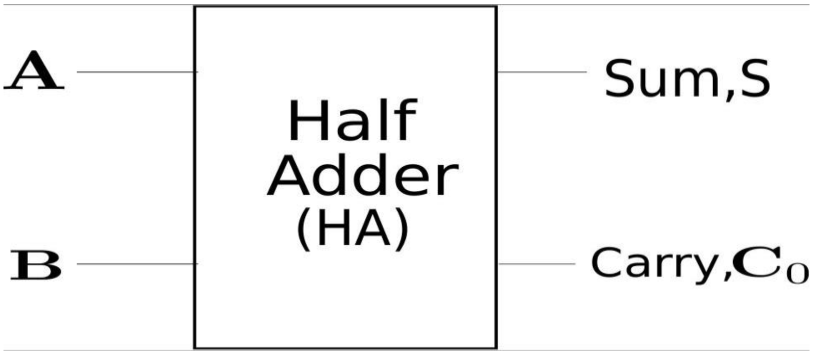


Figure 2 Block Diagram for Half Adder

Table 2 Truth Table for Half-Adder

A	B	S	C_0

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

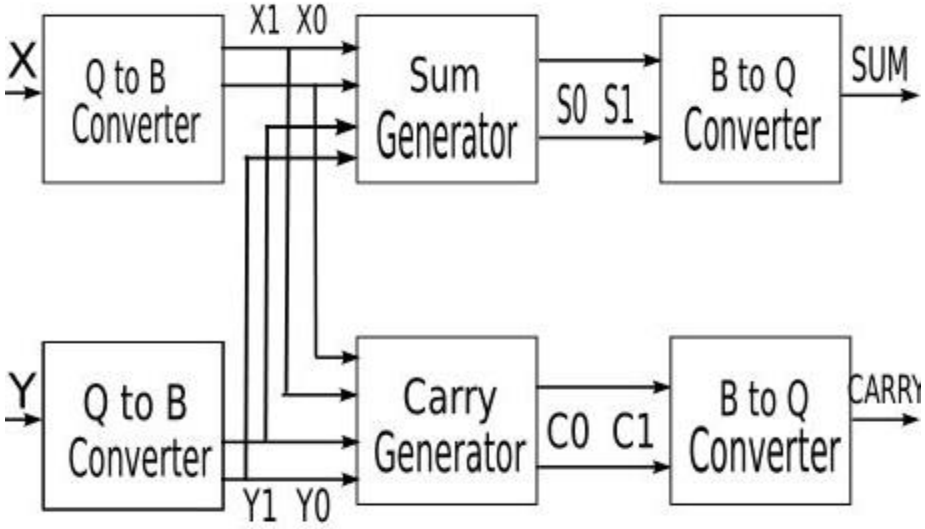


Figure 3 Block diagram of quaternary half adder circumvolution

Where,

- Q = Quaternary Signed Cipher notation
- B = Base-2 Signed Cipher notation
- X and Y are two different inputs where X comprises of X_1 and X_0 and Y comprises of Y_1 and Y_0
- S_0 and S_1 are intervening sum generated
- C_0 and C_1 are Intervening carry generated

4.1.2 FULL ADDER

A full adder could be a combinatory circumvolution that implements the retribution aggregate of 3 bits: *A*, *B* and a direction in, *C*, from the predecessor total. Additionally, as inside the instance of the half-snake rationale and square outlines full viper chart. The aggregate snake deliver the relating aggregate, *S*, and an authorize C_0. As said aforesaid a full snake maybe depicted by 2 half aggregators serial as appeared underneath in Figure three. The aggregate of *A* and *B* range unit sustained to a last half viper, that then adds it to the direction in C (from a precursor accumulation engagement) to get a definitive aggregate *S*. The authorization, C_0, is that the consequences of AND or potentially engagement taken from the direction outs of every half aggregators.[21] The range unit a degree of aggregators inside the writing each at the entryway level and electronic transistor level each giving totally unique upholds.

Boolean Equations:-

$$S = C \oplus \left(A \oplus B\right) \qquad (4)$$

$$C_0 = A \cdot B + C \cdot \left(A \oplus B\right) \qquad (5)$$

With reality table, the entire viper rationale might be authorized. We'll see that the last turnout S is relate XOR between the data sources A and furthermore the half-adder, add up to definite turnout with B and Cin inputs. We tend to take Cout can exclusively be valid if any of the 2 contributions out of the 3 are HIGH.

In this way, we can execute a full snake circumvolution with the help of 2 half adder circumvolutions. At to begin with, half snake will be wont to include *A* and *B* to supply a fractional include and a {second half last half} viper rationale will be wont to add C_{in} to the snake made by the essential half snake to ask a definitive *S* last turnout.[22] On the off chance that any of the half snake rationale deliver a direction, there'll be relate last turnout direction. Thus, C_{out} will be partner OR work of the half-adder trajectory last turnouts. Take a look at the achievement of the aggregate viper circumvolution demonstrated as follows.

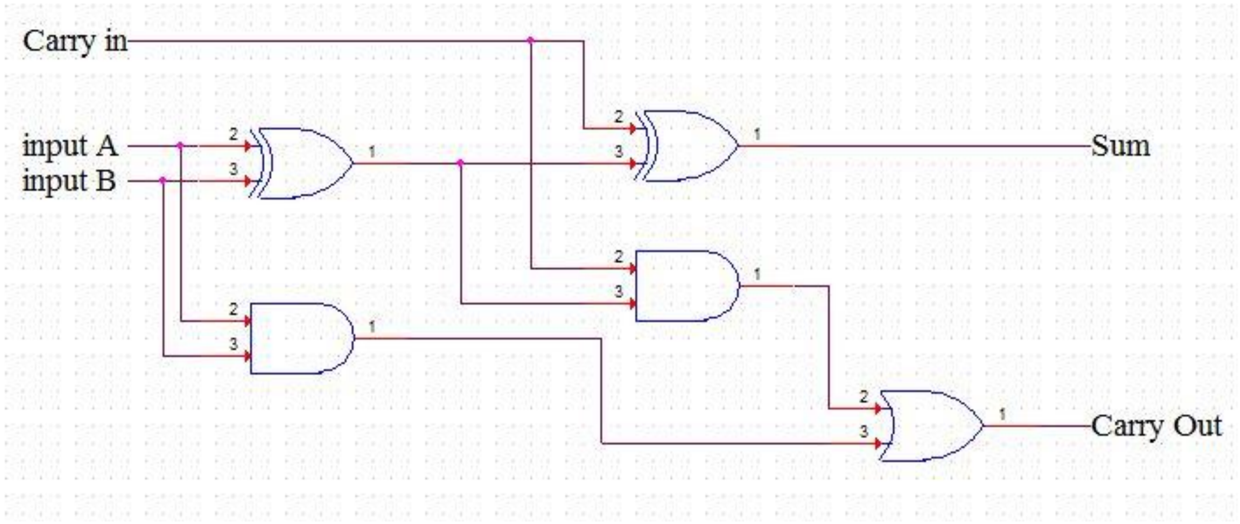


Figure 4 Full adder circumvolution diagram

Table 3 Truth Table for Full Adder

A	**B**	**C**	**S**	**C_0**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

4.1.2.1 RIPPLE CARRY ADDER (RCA)

Half Aggregators will be usual include 2 one piece base-2 cardinal delineations. It's conjointly feasible to make a sensible circumvolution abuse different full aggregators to highlight N-bit base-2 cardinal delineations. Each full viper inputs a Cin, that will be that the Cout of the antecedent adder.[23] This kind of viper might be a Ripple Carry Adder, since each direction bit "swells" to progressive full snake. The essential (and exclusively the main) full viper is additionally supplanted by a half snake. The outline of 4-bit Ripple Carry Adder is appeared next:

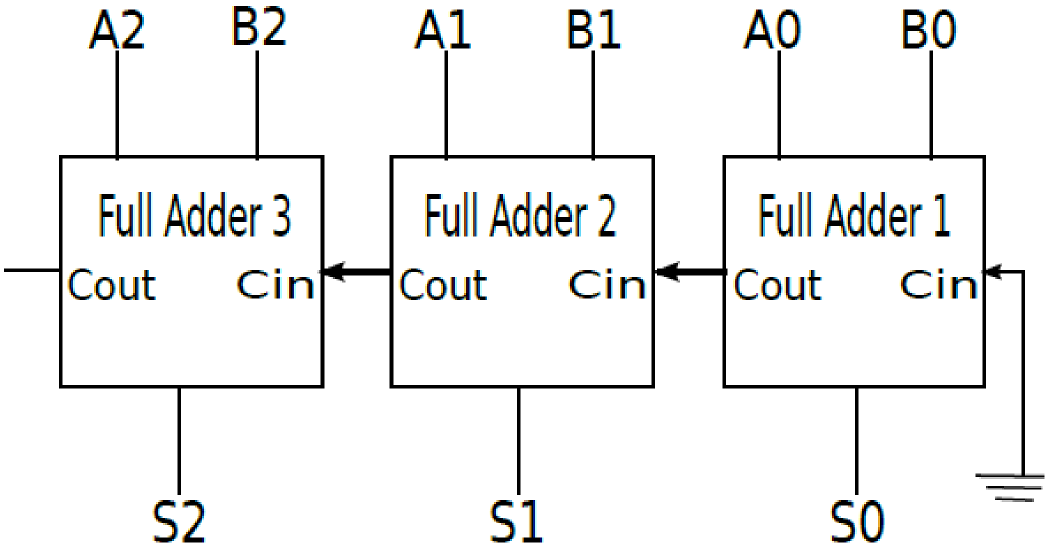


Figure 5 Ripple Carry Adder Block Diagram

$$S_n = A_n \oplus B_n \oplus C_{n-1} \quad (6)$$

$$C_{out} = ((A_n.B_n) + (B_n.C_{n-1}) + (A_n.C_{n-1})) \quad (7)$$

The diagram of Ripple Carry Adder is simple, that grants for fast style time; at the same time, the Ripple Carry Adder is similarly moderate, since each full snake ought to anticipate the direction bit to be computed from the precursor full adder.[24] The entryway deferral will essentially be figured by examination of the aggregate snake circumvolution. Each full viper requirements 3 levels of rationale.

4.1.2.2 CARRY SAVE ADDER (CSA)

A carry-save adder could be a variety of cipher notational adder, utilized in laptop microarchitecture to calculate the aggregation of 3 or aggregation of n-bit cardinal depictions in base-2. It differs from different cipher notational aggregators in this it final turnouts 2 cardinal depictions of identical dimensions because the inputs, one that could be a sequence of partial add bits and an supplementary that could be a sequence of trajectory bits. [25]

The idea of delaying trajectory resolution till the top, or saving carries, is as a result of John Neumann. [25]

Supposing that we've got 2 bits of storage per cipher notation, we will use a redundant base-2 illustration, storing the values zero, 1, 2, or three in every cipher notation position. It's thus obvious that extra} base-2 extent are often added to our trajectory-save result while not overflowing our storage capability.

To place it in an unexpected way, we have a propensity of taking a direction figure documentation from the position to our right side, and spending a direction figure documentation to one side, even as in run of the mill accumulation; however the direction figure documentation we tend to go to one side is that the aftereffects of the predecessor computation and not the present one. In each clock cycle, conveys exclusively should move one stage on, and not n ventures as in run of the mill aggregation. Because signals don't need to move as far, the clock can tick much hustling. [26]

Drawbacks:-

At each stage of a trajectory-save aggregation,

1) We know the result of the aggregation at once.
2) We still do not know whether the result of the aggregation is larger or deficient than a given cardinal depiction (for instance, we do not know whether it is positive or negative).

This latter purpose could be a downside once mistreatment trajectory-save aggregators to implement standard augmentation.

The trajectory-save unit consists of n full aggregators, every of that computes one add and trajectory bit based remarkably alone on the corresponding bits of the 3 input cardinal depictions.

Given the 3 n - bit cardinal depictions a, b, and c, it bring forth a partial add PS and a shift-trajectory sc:

$$ps_i = a_i \oplus b_i \oplus c_i \qquad (8)$$

$$sc_i = (a_i \oplus b_i) \vee (a_i \oplus c_i) \vee (b_i \oplus c_i) \qquad (9)$$

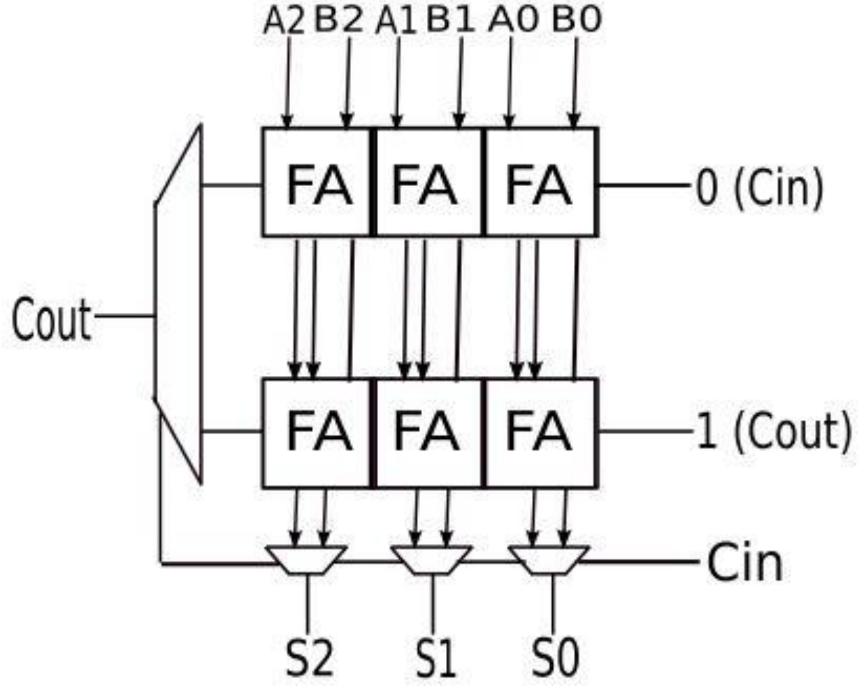


Figure 6 Carry Save Adder Circumvolution

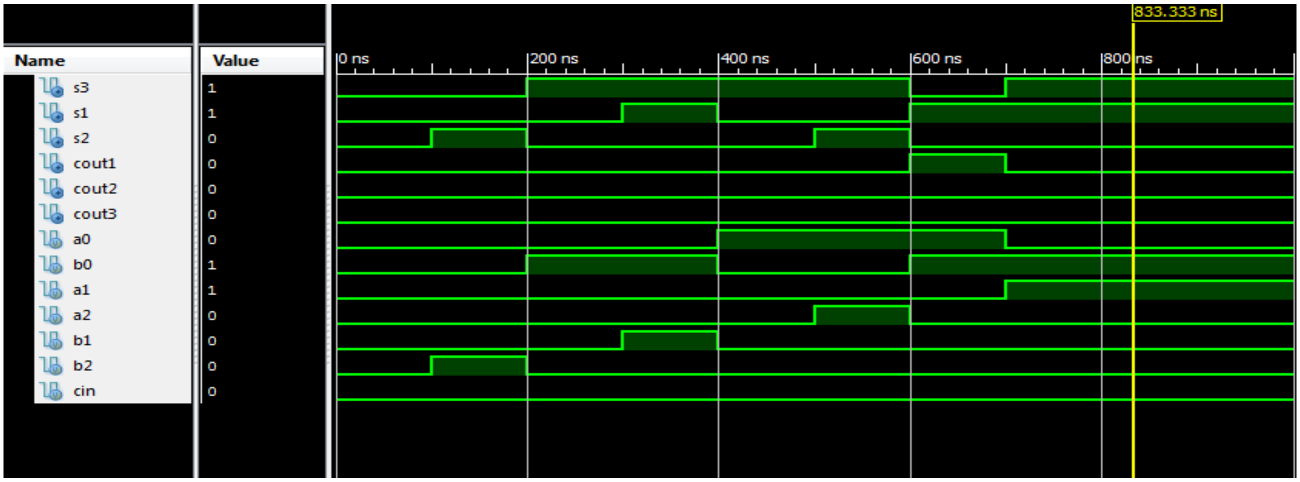
4.2 SIMULATION RESULTS

Successful accomplishment of all the adders discussed above have been implemented successfully and the result of the same has been discussed in the further sub-sections.

4.2.1 RESULTS FOR RIPPLE CARRY ADDER

Figure 7 shows the simulated result for the ripple carry adder verifying the truth table as discussed above.

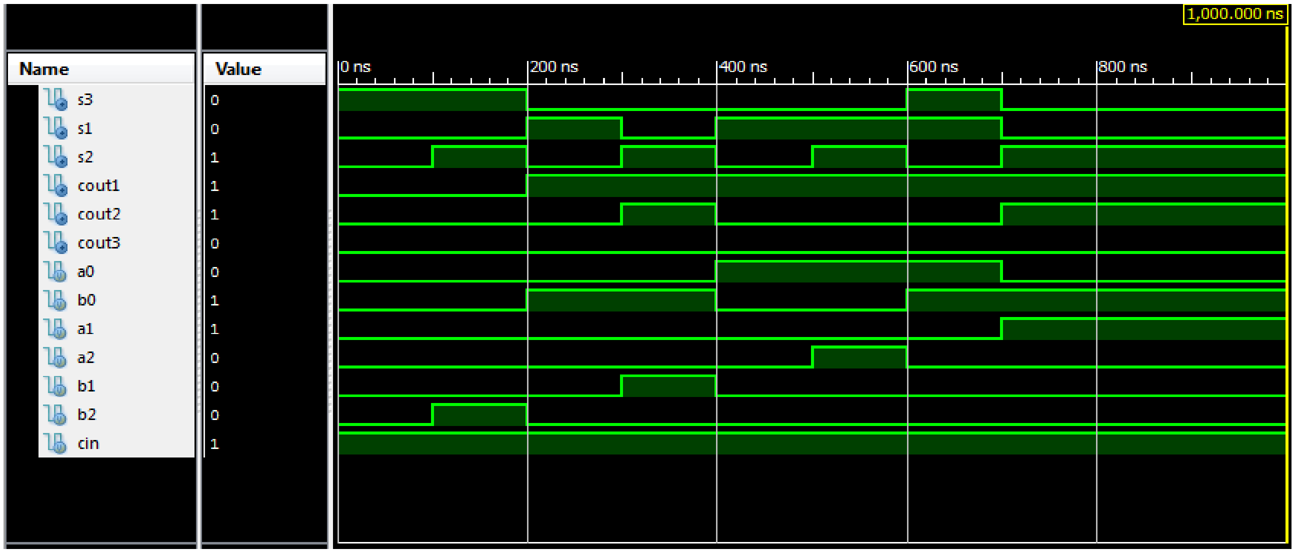
As we can see in figure: A_0, B_0, A_1, B_1, A_2, B_2, C_{in} are the inputs, S_3, S_2, S_1 are the final turnouts and C_{out1}, C_{out2}, C_{out3} are the trajectory final turnouts.



Name	Value
s3	1
s1	1
s2	0
cout1	0
cout2	0
cout3	0
a0	0
b0	1
a1	1
a2	0
b1	0
b2	0
cin	0

Figure 7 Simulation Result of Ripple Carry Adder

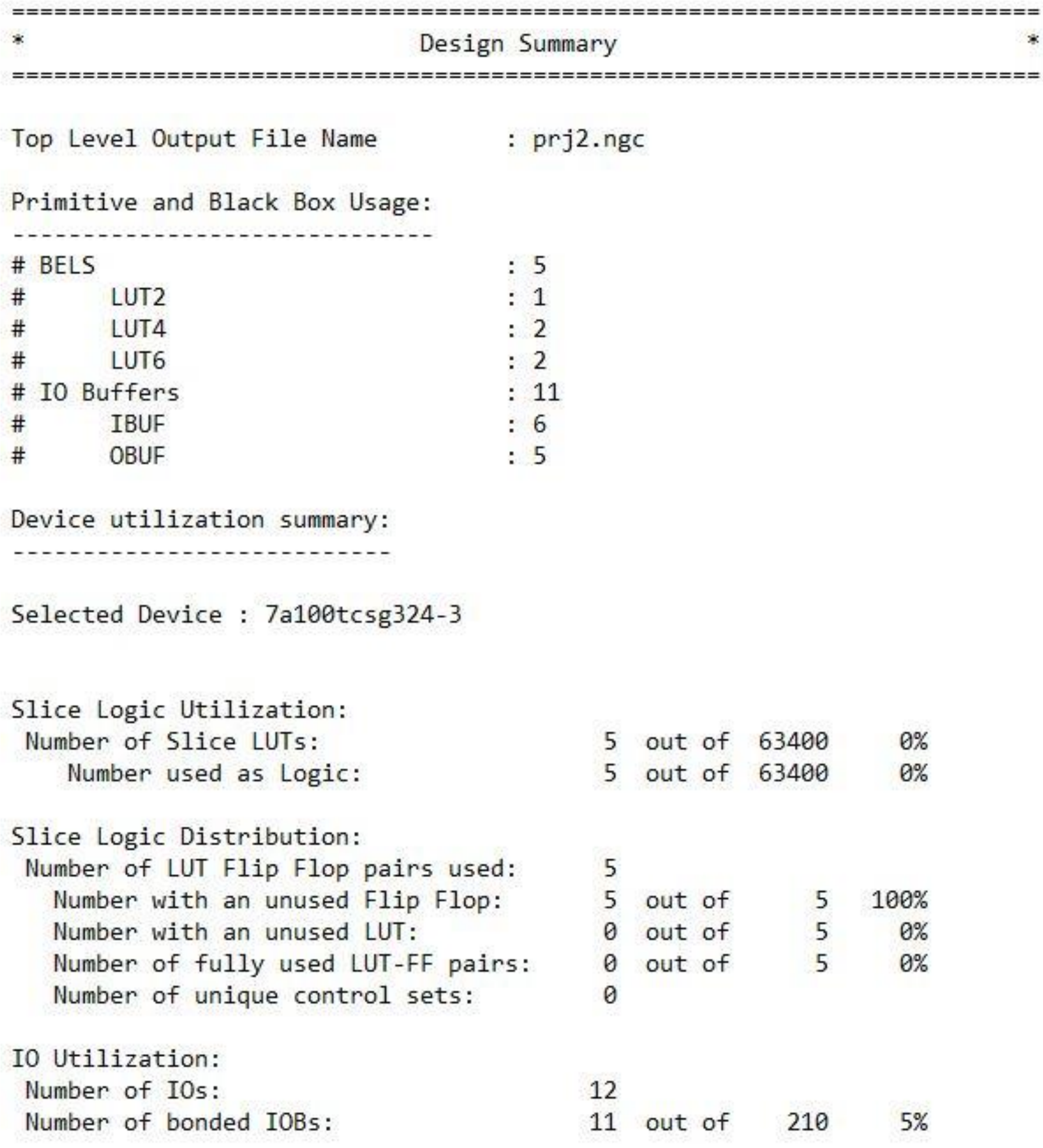
Figure 8 shows simulated result for the ripple carry subtractor. This result has been obtained by changing the C_{in} value from 0 to 1 in the testbench of the Xilinx code.



Name	Value
s3	0
s1	0
s2	1
cout1	1
cout2	1
cout3	0
a0	0
b0	1
a1	1
a2	0
b1	0
b2	0
cin	1

Figure 8 Simulation Result of Ripple Carry Subtractor

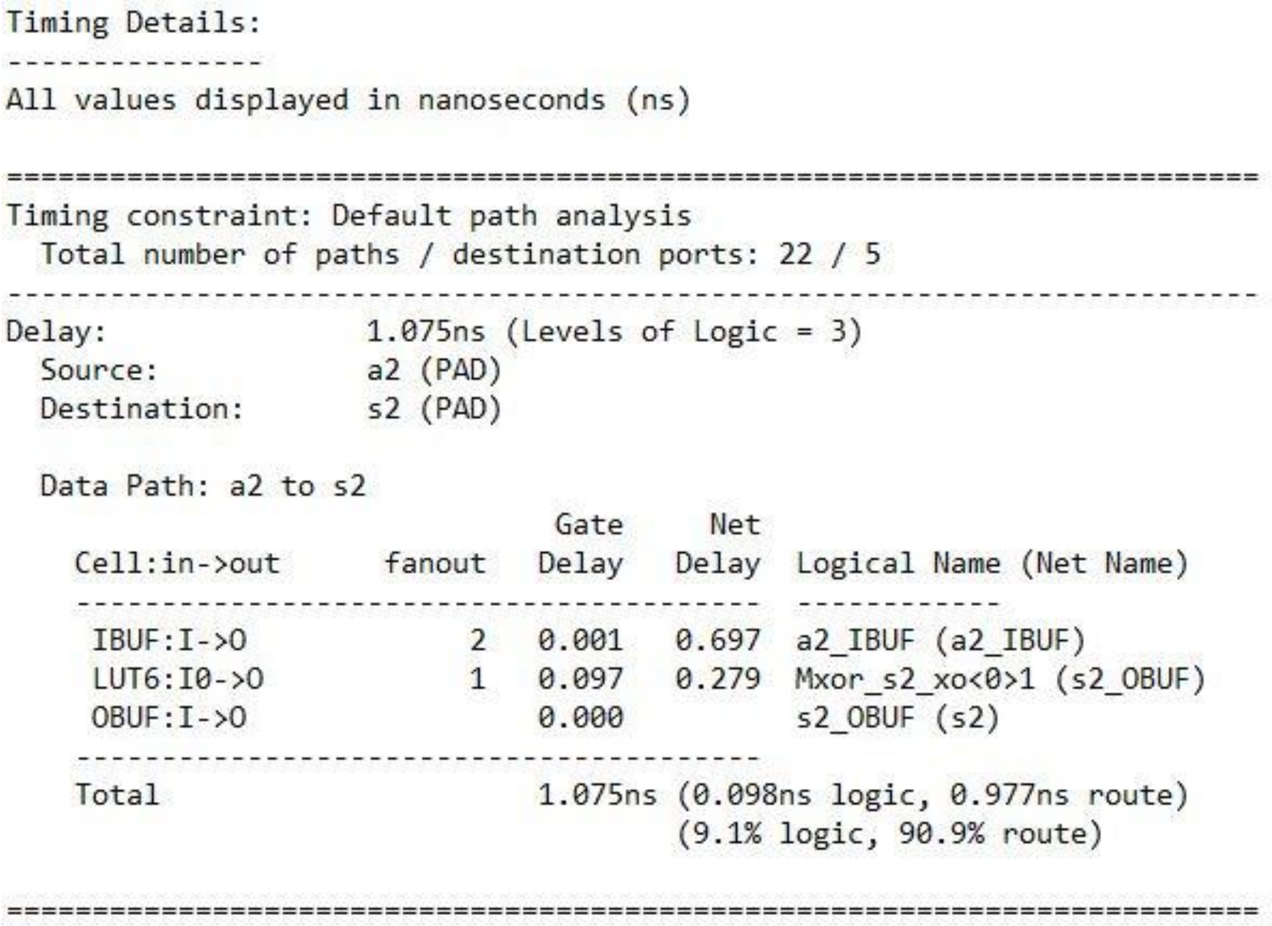
Figure 9 shows the deign summary of the ripple carry adder as well subtractor. This synthesis report shows the total number of LUT's and Slices used during the simulation of the RCA circuit which we will further use in the result comparison later.



```
=========================================================================
*                          Design Summary                               *
=========================================================================

Top Level Output File Name         : prj2.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 5
#      LUT2                        : 1
#      LUT4                        : 2
#      LUT6                        : 2
# IO Buffers                       : 11
#      IBUF                        : 6
#      OBUF                        : 5

Device utilization summary:
---------------------------

Selected Device : 7a100tcsg324-3


Slice Logic Utilization:
 Number of Slice LUTs:                    5  out of  63400     0%
    Number used as Logic:                 5  out of  63400     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:      5
   Number with an unused Flip Flop:       5  out of      5   100%
   Number with an unused LUT:             0  out of      5     0%
   Number of fully used LUT-FF pairs:     0  out of      5     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                          12
 Number of bonded IOBs:                  11  out of    210     5%
```

Figure 9 Synthesis Report Ripple Carry Adder

Figure 10 shows the timing details for the ripple carry adder as well as subtractor. Here we consider three different time lags which is the total delay, the logic delay and the route delay of the circuit. These time lag values will be used for comparison purpose later.



```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 22 / 5
-------------------------------------------------------------------------
Delay:               1.075ns (Levels of Logic = 3)
  Source:            a2 (PAD)
  Destination:       s2 (PAD)

  Data Path: a2 to s2
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   0.001   0.697  a2_IBUF (a2_IBUF)
     LUT6:I0->O            1   0.097   0.279  Mxor_s2_xo<0>1 (s2_OBUF)
     OBUF:I->O                 0.000          s2_OBUF (s2)
    ----------------------------------------
    Total                      1.075ns (0.098ns logic, 0.977ns route)
                                       (9.1% logic, 90.9% route)

=========================================================================
```

Figure 10 Time Lag for Ripple Carry Adder

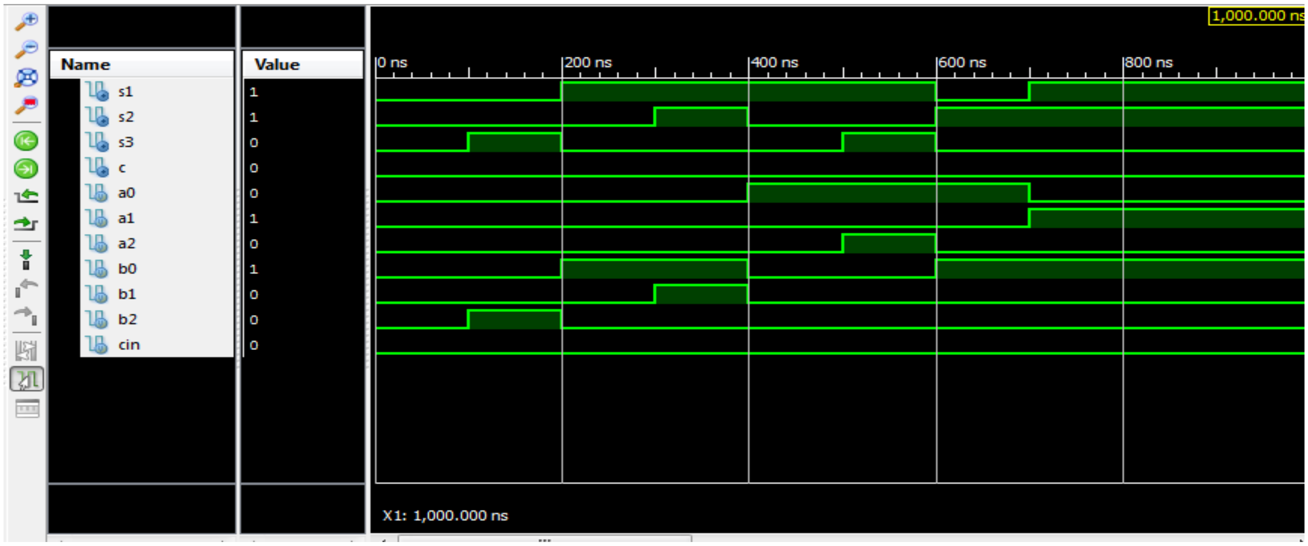
In table 4, different values obtained from the synthesis report as well as the time lag has been compared with various other results obtained after studying various research papers.

Table 4 Comparison Table for Ripple Carry Adder

Adder Type	Method	LUT	Slices	Gates	Delay (ns)	Logic Delay (ns)	Route Delay (ns)
Ripple Carry Adder	A.DivyaPrabha. *et al.* [10]	11	4	15	44.366	11.28	45.393
	A.Avizinis *et al.* [11]	8	4	15	1.545	0.94	0.98
	Our Work	5	5	15	1.075	0.98	0.977

4.2.2 RESULTS FOR CARRY SAVE ADDER

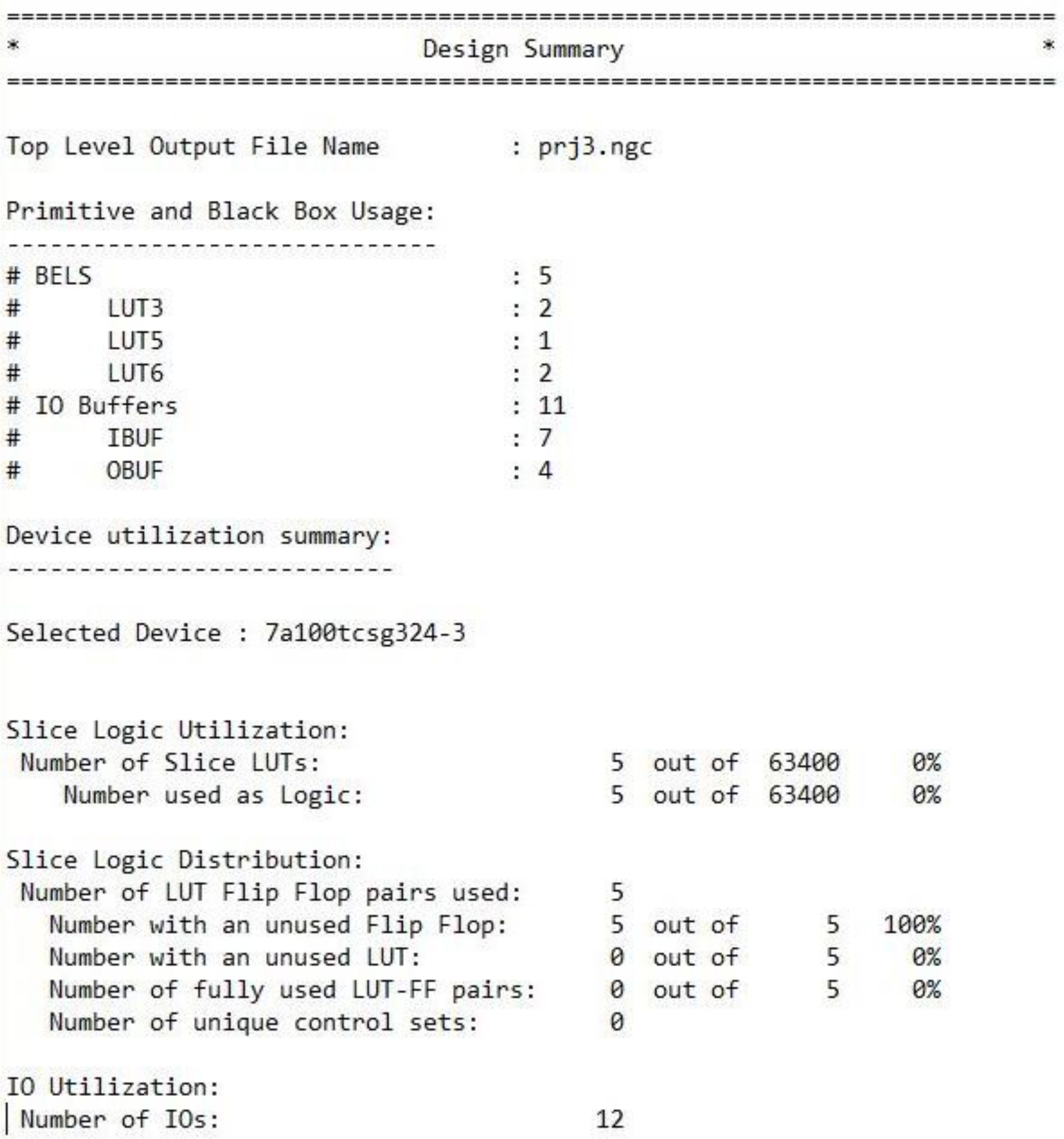
Figure 11 shows the simulated result for the carry save adder verifying the truth table as discussed above. As we can see in figure: A_0, B_0, A_1, B_1, A_2, B_2, C_{in} are the inputs, S_3, S_2, S_1 are the final turnouts and C is the final trajectory final turnout.



Name	Value
s1	1
s2	1
s3	0
c	0
a0	0
a1	1
a2	0
b0	1
b1	0
b2	0
cin	0

Figure 11 Simulation Result of Carry Save Adder

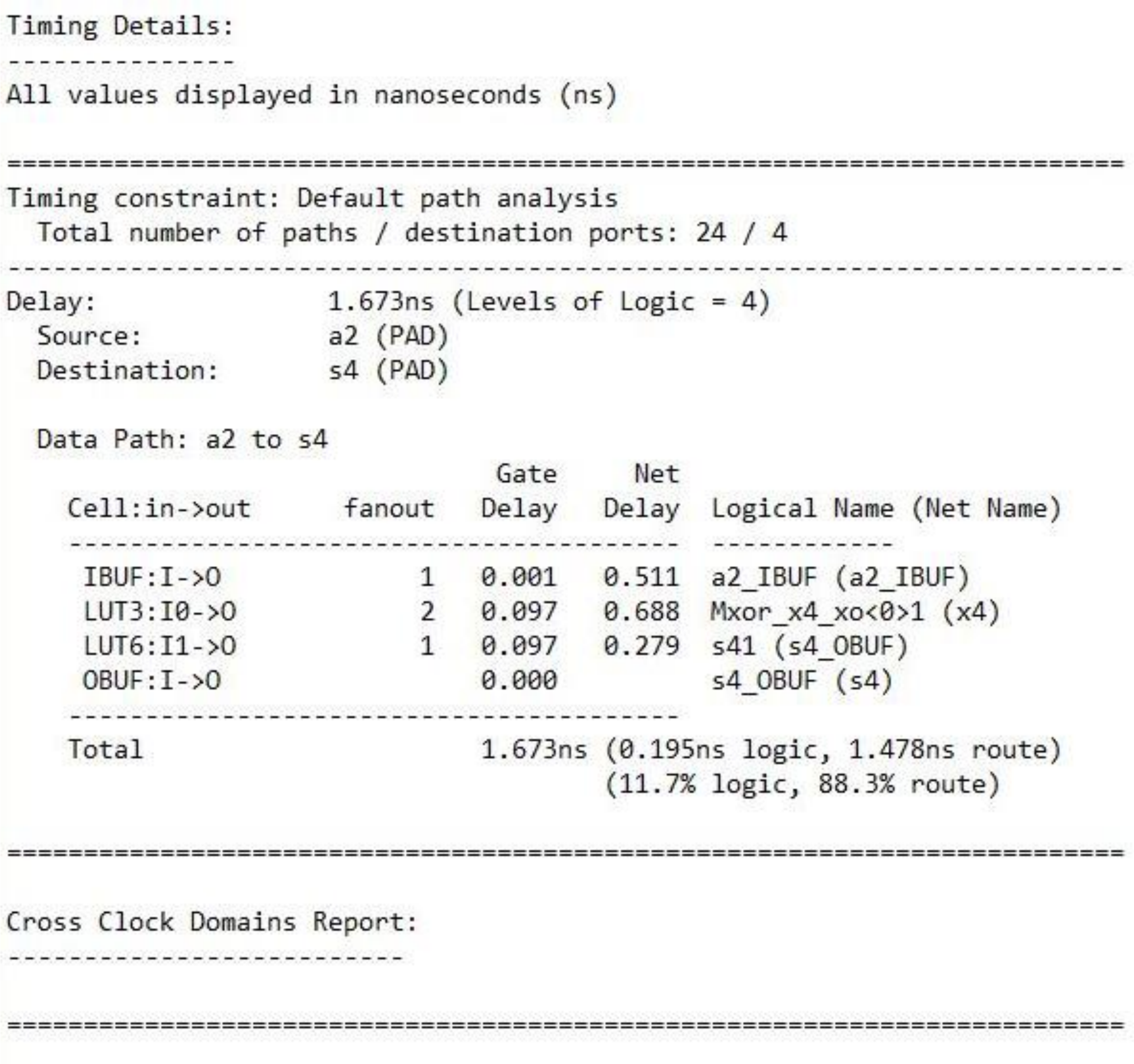
Figure 12 shows the deign summary of the carry save adder. This synthesis report shows the total number of LUT's and Slices used during the simulation of the CSA circuit which we will further used in the result comparison later.



```
=========================================================================
*                            Design Summary                             *
=========================================================================

Top Level Output File Name         : prj3.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 5
#      LUT3                        : 2
#      LUT5                        : 1
#      LUT6                        : 2
# IO Buffers                       : 11
#      IBUF                        : 7
#      OBUF                        : 4

Device utilization summary:
---------------------------

Selected Device : 7a100tcsg324-3


Slice Logic Utilization:
 Number of Slice LUTs:                    5  out of  63400     0%
    Number used as Logic:                 5  out of  63400     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:      5
   Number with an unused Flip Flop:       5  out of      5   100%
   Number with an unused LUT:             0  out of      5     0%
   Number of fully used LUT-FF pairs:     0  out of      5     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                          12
```

Figure 12 Synthesis Report Carry Save Adder

Figure 13 shows the timing details for the carry save adder. Here we consider three different time lags which is the total delay, the logic delay and the route delay of the circuit. These time lag values will be used for comparison purpose later.



```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 24 / 4
-------------------------------------------------------------------------
Delay:               1.673ns (Levels of Logic = 4)
  Source:            a2 (PAD)
  Destination:       s4 (PAD)

  Data Path: a2 to s4
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             1   0.001   0.511  a2_IBUF (a2_IBUF)
     LUT3:I0->O            2   0.097   0.688  Mxor_x4_xo<0>1 (x4)
     LUT6:I1->O            1   0.097   0.279  s41 (s4_OBUF)
     OBUF:I->O                 0.000          s4_OBUF (s4)
    ----------------------------------------
    Total                      1.673ns (0.195ns logic, 1.478ns route)
                                       (11.7% logic, 88.3% route)

=========================================================================

Cross Clock Domains Report:
--------------------------

=========================================================================
```

Figure 13 Time Lag for Carry Save Adder

In table 5, different values obtained from the synthesis report as well as the time lag has been compared with various other results obtained after studying various research papers.

Table 5 Comparison Table for Carry Save Adder

Adder Type	Method	LUT	Slices	Gates	Delay (ns)	Logic Delay (ns)	Route Delay (ns)
Ripple Carry Adder	Maroju SaiKumar *et al.* 2013[29]	13	9	20	1.433	0.289	1.144
	Aritra Mitra *et al.* 2015[30]	17	11	20	3.09	1.349	1.741
	Our Work	5	5	20	1.673	0.195	1.478

CHAPTER 5: PROPOSED WORK

In this chapter we are discussing about how to generate the sum final turnout without generation of the trajectory. Therefore, we have contended trajectory quaternary addition and have successfully implemented the contended work which is discussed in the upcoming sub – section.

5.1 CARRY FREE ADDITION

The vary of QSD cardinal depictions is from -3 to +3, and once the 2 QSD cardinal depictions square measure superimposed the result extents from -6 to +6. The final turnout of all the potential mixtures of 2 cardinal depictions is illustrated in Table a pair of. If the vary is exceeded by the decimal variety then there's a demand of 1 aggregation QSD cipher notation. Within the aggregation of 2 cipher notation QSD result, LSB is characterised by the total bit and mutual savings bank bit is characterised by trajectory. One QSD portrayals are often accustomed represent an equivalent decimal variety. The quantity is chosen so as to stop the ripple of the trajectory.[27]

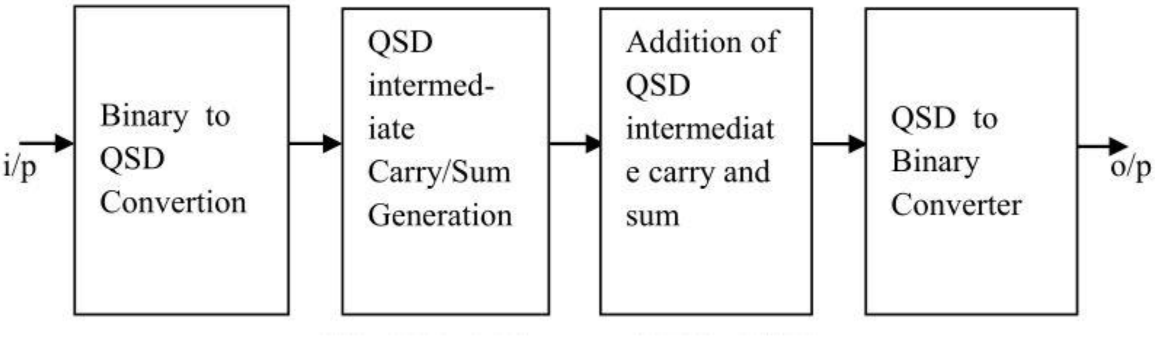

i/p
Binary to QSD Convertion
QSD intermed-iate Carry/Sum Generation
Addition of QSD intermediat e carry and sum
QSD to Binary Converter
o/p

Figure 14 Block Diagram of Carry Free QSD Addition

5.2 RULES FOR CARRY FREE ADDTION

In the aggregation of 2 cipher notation QSD result, LSB is characterised by the add bit and MSB bit is characterised by trajectory. Over one QSD portrayals are often wont to represent an equivalent decimal extent. The amount is chosen so as to stop the ripple of the trajectory. The aggregation of {the 2} QSD cardinal depictions while not trajectory are often worn out the subsequent two steps [25]:

In step 1, trajectory and add that area unit intervening is generated by the adder from the input cipher notations.

In step 2, adder adds the (intervening) of current cipher notation with the trajectory (intervening) of lower figure.

The ripple of trajectory are often removed by in engagement the subsequent 2 rules in QSD aggregation [27]:

Rule 1 states that the greatness of the aggregate (mediating) must be scarcely than or equivalent to 2 i.e., it ought to be in the degree of - 2 to +2.

Rule 2 states that the greatness of the direction (interceding) must be scarcely than or equivalent to 1 i.e., it ought to be in the degree of - 1 to +1.

From the primary step we have a tendency to see that, the intervening total and therefore the Intervening carry is obtained from the QSD adder that lies within the extent of -6 to +6 as mentioned within the on top of rules. However mistreatment redundancy attribute of QSD cardinal depictions, 655 is chosen that satisfies the on top of explicit rules. once the intervening total whose vary is from -2 to +2 as obtained within the second step of QSD adder is superimposed with the lower figure Intervening carry whose vary is -1 to +1, the final turnout of the aggregation can't be bigger than three i.e., it'll be within the vary of -3 to +3 and therefore, there's no aggregation demand of trajectory because the aggregation final turnout during this vary will be characterized by a 1 cipher notation QSD extent [28].

Table 6 Intervening sum and carry from -6 to +6

Sum	Possible QSD Portrayals	QSD Cardinal depiction
-6	$\overline{12}, \bar{2}2$	$\overline{12}$

-5	$\overline{11}, \bar{2}3$	$\overline{11}$
-4	$\bar{1}0$	$\bar{1}0$
-3	$\bar{1}1, 0\bar{3}$	$\bar{1}1$
-2	$\bar{1}2, 0\bar{2}$	$0\bar{2}$
-1	$\bar{1}3, 0\bar{1}$	$0\bar{1}$
0	0 0	00
1	$01, 1\bar{3}$	01
2	$02, 1\bar{2}$	02
3	$03, 1\bar{1}$	$1\bar{1}$
4	1 0	10
5	$11, 2\bar{3}$	11
6	$12, 2\bar{2}$	12

Table 7 Alteration between the inputs and final turnouts of the Intervening carry and Intervening sum

QSD		**BSD**		**DECIMAL**	**QSD**		**BSD**	
A_i	**B_i**	**A_i**	**B_i**	$\sum$	**C_i**	**S_i**	**C_i**	**S_i**
3	3	011	011	6	1	2	001	010
3	2	011	010	5	1	1	001	001
2	3	010	011	5	1	1	001	001
3	1	011	001	4	1	0	001	000

1	3	001	011	4	1	0	001	000
2	2	010	010	4	1	0	001	000
1	2	001	010	3	1	-1	001	111
2	1	010	001	3	1	-1	001	111
3	0	011	000	3	1	-1	001	111
0	3	000	011	3	1	-1	001	111
1	1	001	001	2	0	2	000	010
0	2	000	010	2	0	2	000	010
2	0	010	000	2	0	2	000	010
3	-1	011	111	2	0	2	000	010
-1	3	111	011	2	0	2	000	010
0	1	000	001	1	0	1	000	001
1	0	001	000	1	0	1	000	001
2	-1	010	111	1	0	1	000	001
-1	2	111	010	1	0	1	000	001
3	-2	011	110	1	0	1	000	001
-2	3	110	011	1	0	1	000	001
0	0	000	000	0	0	0	000	000
1	-1	001	111	0	0	0	000	000
-1	1	111	001	0	0	0	000	000
2	-2	010	110	0	0	0	000	000
-2	2	110	010	0	0	0	000	000
-3	3	101	011	0	0	0	000	000

3	-3	011	101	0	0	0	000	000
0	-1	000	111	-1	0	-1	000	111
-1	0	111	000	-1	0	-1	000	111
-2	1	110	001	-1	0	-1	000	111
1	-2	001	110	-1	0	-1	000	111
-3	2	101	010	-1	0	-1	000	111
2	-3	010	101	-1	0	-1	000	111
-1	-1	111	111	-2	0	-2	000	110
0	-2	000	110	-2	0	-2	000	110
-2	0	110	000	-2	0	-2	000	110
-3	1	101	001	-2	0	-2	000	110
1	-3	001	101	-2	0	-2	000	110
-1	-2	111	110	-3	-1	1	111	001
-2	-1	110	111	-3	-1	1	111	001
-3	0	101	000	-3	-1	1	111	001
0	-3	000	101	-3	-1	1	111	001
-3	-1	101	111	-4	-1	0	111	000
1	-3	111	101	-4	-1	0	111	000
-2	-2	110	110	-4	-1	0	111	000
3	-2	101	110	5	1	1	111	111
-2	-3	110	101	5	1	1	111	111
3	3	101	101	6	1	2	111	110

We see that within the Table two the primary column shows the multiple portrayals of some cardinal depictions, however solely of those who satisfies the on top of nominative rules an elect. Currently once we see the last column of the Table one that shows the chosen intervening total and trajectory of the coded QSD variety. An illustration is given below to administer clarity of the aggregation method:[29]

Let us enforce QSD aggregation of two cardinal depictions:

A = 108 and *B*=-232 (positive and negative respectively).

Decimal cardinal depiction is first converted into its equivalent QSD portrayal:

$$(108)_{10} = 1\times 4^3 + 2\times 4^2 + 3\times 4^1 + 0\times 4^0 = (1230)_{QSD}$$

$$(232)_{10} = 3\times 4^3 + 3\times 4^2 + 2\times 4^1 + 0\times 4^0 = (3320)_{QSD}$$

$$Hence, (-232)_{10} = (3320)_{QSD}$$

When we add the two QSD cardinal depictions the aggregation can be done in the following way:

Table 8 Addition of two QSD cardinal depictions

A=108	$2\bar{2}30$
B=-232	$\bar{3}\bar{3}20$
Decimal Sum	-1 -5 5 0
IC	0 -1 1 0
IS	-1-1 1 0
S	$\bar{2}010$
C	0

The final turnout of the adder of the aggregation of 108 and -232 in QSD mathematical notation is (2’010)QSD whose decimal equivalent is (-124)10 and trajectory final turnout is zero. From this illustration it's clear that the QSD adder style method carries the 2 stage aggregation.

By scaling the 2 cipher notations into intervening add and Intervening carry specified the ordinal intervening add and therefore the (n-1)th. Confining the portrayal of the QSD variety in keeping with the foundations as outlined on top of the terminal aggregation can don't have any trajectory within the finish.[29]

5.3 EQUATIONS AND CIRCUMVOLUTION DIAGRAM

On considering the information side, the numbers to be added A_i is portrayed by 3 variable contribution as A_2, A_1, A_0 and the cardinal delineation B_i is portrayed by 3 variable contribution as B_2, B_1, B_0. Whereas, on thought of the last turnout side, the Intervening convey IC is portrayed by IC_2, IC_1, IC_0 and the mediating entirety is portrayed by IS_2, IS_1, IS_0. The six variable expressions for Intervening convey and interceding entirety as far as sources of info (A_2, A_1, A_0, B_2, B_1 and B_0). So we get the six last turnout expressions for IC_2, IC_1, IC_0, IS_2, IS_1 and IS_0.The rationale conditions for limited equipment acknowledgment for creating the Intervening convey and mediating aggregate are determined utilizing 6 variable K-outline.[30]

The min terms for the Intervening carry (IC_2, IC_1, IC_0) are:

$$IS_0 = a_0 \oplus b_0 \tag{10}$$

$$IS_1 = \left(a_0.b_0\right) \oplus a_1 \oplus b_1 \tag{11}$$

$$\begin{aligned} IS_2 = {} & \left(IS_0.\left(a_1 \oplus b_1\right)\right) + \left(\overline{a}_1.\overline{b}_0.b_2\right) + \left(\left(a_0.b_0\right).\left(\overline{a}_1.\overline{b}_1\right).\left(a_2 + b_2\right)\right) \\ & + \left(a_0.b_0.a_1.b_1.a_2.b_2\right) \end{aligned} \tag{12}$$

The min terms for Intervening carry are:

$$IC_2 = IC_1 = (a_2.b_2).(\overline{a_0.b_0.a_1.b_1}) + (\overline{a_1 + b_1}).(a_2.\overline{b}_0 + \overline{a}_0.b_2) \tag{13}$$

$$IC_0 = IC_2 + (\overline{a_2.b_2}).(a_1.b_1 + b_1.b_0 + a_1.b_0 + a_0.b_1 + a_1.a_0) \tag{14}$$

The end amazingly last turnout is the entirety which is direction complimentary produced from the above definite turnouts. In this manner it has 6 sources of info and 3 last turnout bits.

$$S_0 = IC_0 \, . \, \overline{IS_0} + \overline{IC_0} \, . \, IS_0 \tag{15}$$

$$S_1 = IC_1 \oplus IS_1 \oplus \left(IC_0 \, . \, IS_0 \right) \tag{16}$$

$$S_2 = IC_2 \oplus IS_2 \oplus \Big(\left(IC_1 \, . \, IS_1 \right) + \left((IC_1 + IS_1) . \left(IC_0 \, . \, IS_0 \right) \right) \Big) \tag{17}$$

The circumvolution is obtained from the above equations and we have deigned the circumvolution using PSPICE software.

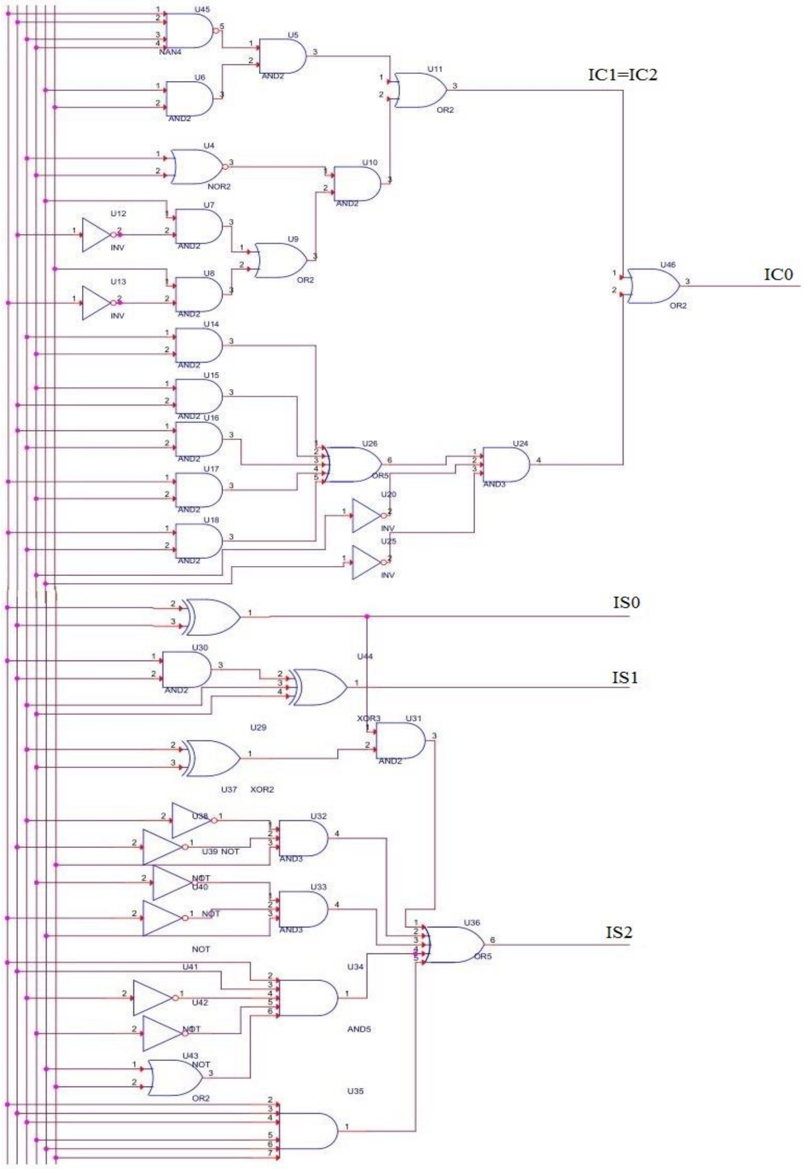


Figure 15 Gates combinational circuit diagram for single digit QSD adder cell

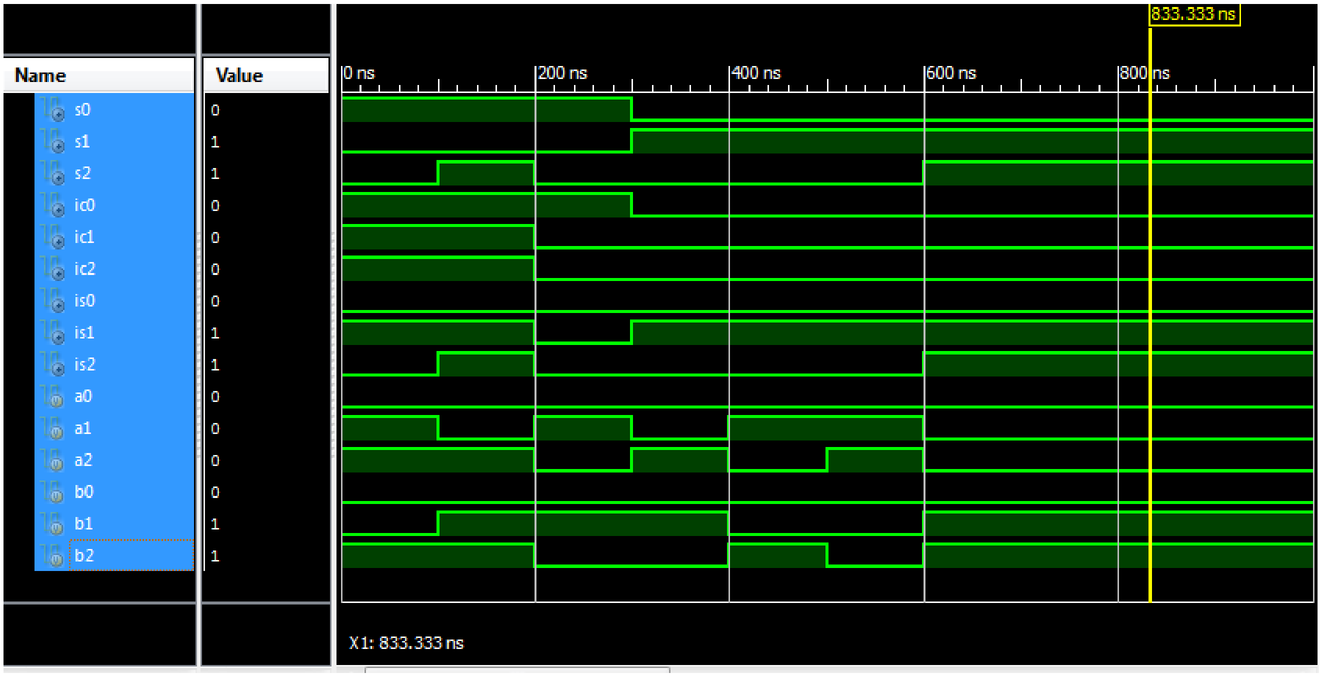
5.4 SIMULATION RESULTS

Successful accomplishment of all the carry free adder discussed above have been implemented successfully and the result of the same has been discussed in the further sub-section.

5.4.1 RESULTS FOR CARRY FREE ADDER

Figure 16 shows the simulated diagram for the carry free adder verifying the truth table as discussed above. Different variables seen in the diagram are detailed below:

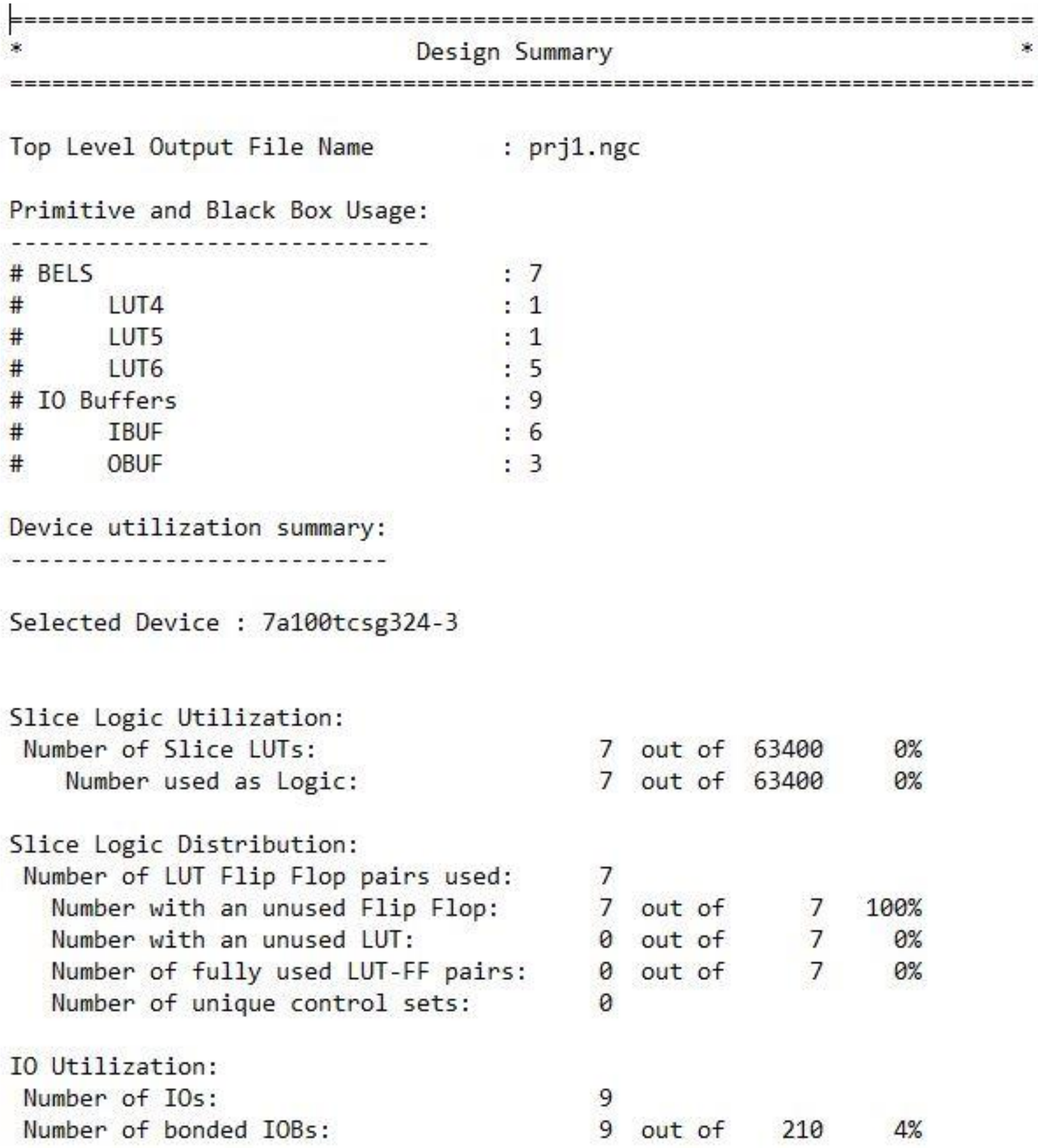
As we can see in figure: A_0, B_0, A_1, B_1, A_2, B_2 are the inputs, S_3, S_2, S_1 are the final turnouts, IC_0, IC_1, IC_2 are the Intervening carry final turnouts and IS_3, IS_2, IS_1 are the intervening sum final turnouts.



Name	Value
s0	0
s1	1
s2	1
ic0	0
ic1	0
ic2	0
is0	0
is1	1
is2	1
a0	0
a1	0
a2	0
b0	0
b1	1
b2	1

Figure 16 Simulation Result of QSD Adder

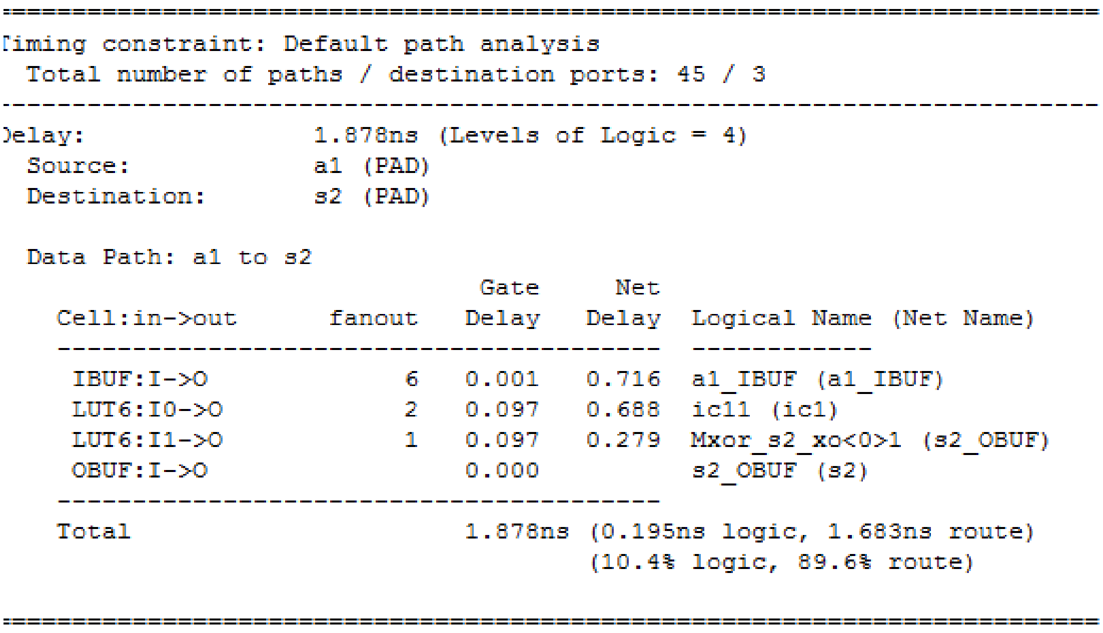
Figure 17 shows the deign summary of the carry free adder. This synthesis report shows the total number of LUT's and Slices used during the simulation of the CFA circuit which we will further used in the result comparison later



```
=========================================================================
*                          Design Summary                               *
=========================================================================

Top Level Output File Name         : prj1.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 7
#      LUT4                        : 1
#      LUT5                        : 1
#      LUT6                        : 5
# IO Buffers                       : 9
#      IBUF                        : 6
#      OBUF                        : 3

Device utilization summary:
---------------------------

Selected Device : 7a100tcsg324-3


Slice Logic Utilization:
 Number of Slice LUTs:                    7  out of  63400     0%
    Number used as Logic:                 7  out of  63400     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:      7
   Number with an unused Flip Flop:       7  out of      7   100%
   Number with an unused LUT:             0  out of      7     0%
   Number of fully used LUT-FF pairs:     0  out of      7     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                           9
 Number of bonded IOBs:                   9  out of    210     4%
```

Figure 17 Synthesis Report of Carry Free Adder

Figure 18 shows the timing details for the carry free adder. Here we consider three different time lags which is the total delay, the logic delay and the route delay of the circuit. These time lag values will be used for comparison purpose later.



```
================================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 45 / 3
--------------------------------------------------------------------------------
Delay:               1.878ns (Levels of Logic = 4)
  Source:            a1 (PAD)
  Destination:       s2 (PAD)

  Data Path: a1 to s2
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             6   0.001   0.716  a1_IBUF (a1_IBUF)
     LUT6:I0->O            2   0.097   0.688  ic11 (ic1)
     LUT6:I1->O            1   0.097   0.279  Mxor_s2_xo<0>1 (s2_OBUF)
     OBUF:I->O                 0.000          s2_OBUF (s2)
    ----------------------------------------
    Total                      1.878ns (0.195ns logic, 1.683ns route)
                                       (10.4% logic, 89.6% route)

================================================================================
```

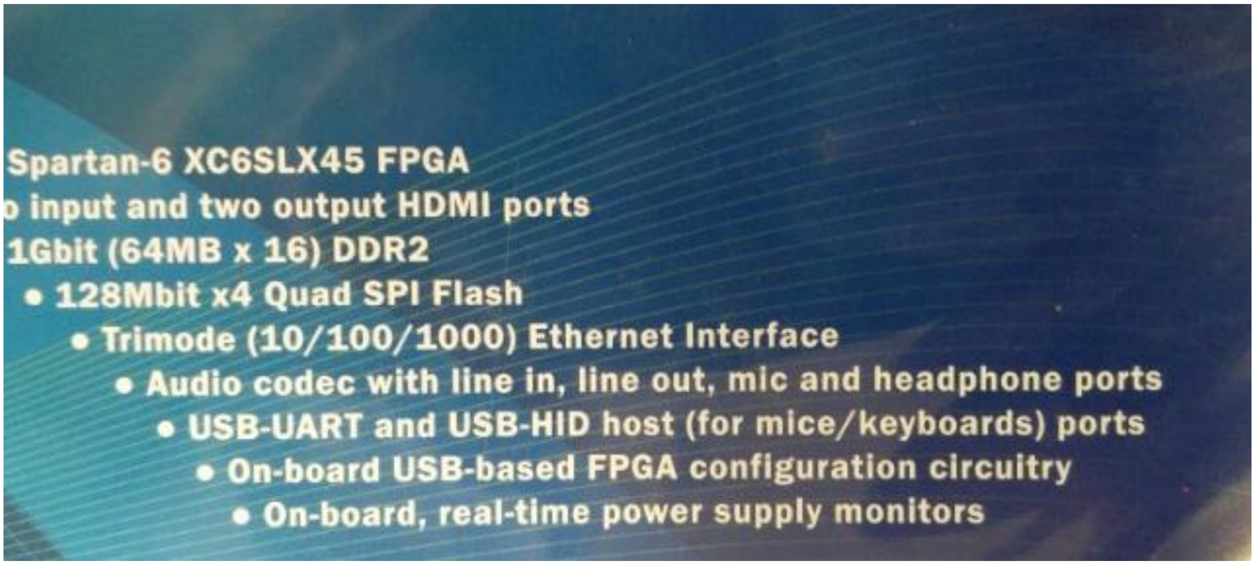
Figure 18 Time lag for Carry Free Addition

In table 9, different values obtained from the synthesis report as well as the time lag has been compared with various other results obtained after studying various research papers.

Table 9 Comparison Table for Carry Free Adder

Adder Type	Method	LUT	Slices	Gates	Delay (ns)	Logic Delay (ns)	Route Delay (ns)
Carry Free Addition	R. BHUKYA *et al.* 2014[9]	17	30	38	2	0.333	1.893
	S.Jakeer Hussain *et al* 2014[12]	28	47	38	2	0.268	2.257
	Our Work	9	7	38	1.878	0.195	1.638

The Simulation of QSD subtraction which is written in Verilog HDL is done in Xilinx Software 14.7.We have burnt the circumvolution on the FPGA successfully as shown below:



Spartan-6 XC6SLX45 FPGA

o input and two output HDMI ports

1Gbit (64MB x 16) DDR2

- 128Mbit x4 Quad SPI Flash
- Trimode (10/100/1000) Ethernet Interface
- Audio codec with line in, line out, mic and headphone ports
- USB-UART and USB-HID host (for mice/keyboards) ports
- On-board USB-based FPGA configuration circuitry
- On-board, real-time power supply monitors

Figure 19 FPGA version details

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

We conclude that the time lag for the Ripple Carry Adder circumvolution comes out to be 1.075ns whereas the time lag of Carry Save Adder circumvolution comes out to be 1.673ns.

The suggested design of QSD Carry Free adder circumvolution generates the time lag of 1.878ns. These circumvolutions expend scarcely vitality, indicates considerable power productivity prompting better enforcement. We see that the, slack of the proposed configuration is scarcely. Therefore of which this outline is proper to be connected for the achievement of an impressive authorization multiprocessor which may comprise of many handling components.

6.2 FUTURE WORK

As we have successfully implemented three different types of adder circumvolutions using various combinatory gates we now have enough knowledge to further take this work on advance level which is augmentation of the quaternary signed cipher notations using aggregators as the basic building block in the process of building up the multipliers.

REFERENCES:

[1] S.Dubey, R.Rani "VLSI Accomplishment of Fast Aggregation using Quaternary Signed Digit Cardinal depiction Methodology" IEEE International Conference on Emerging Trends in Taking into account, Communication and Nanotechnology (ICECCN), pp. 654-659, 2013.

[2] A.A.S. Awwal, J.Ahmed, "Fast Trajectory complimentary Adder Design Using QSD Cardinal depiction Methodology". IEEE, CH3306-8/93/0000-1085, Pp. 1085-1088, 1993.

[3] B.Parhami, "Trajectory-Complimentary Aggregation of Recoded Base-2 Signed- Cipher notation Cardinal depictions", IEEE Transactions on Computers, Vol. 37, No. 11, pp. 1470-1476, November 1988.

[4] P.S. Kamble & S.M. Choudhary, "Review of VHDL Accomplishment of Quaternary Signed Aggregator methodology" IEEE 1993 national aerospace and electronic Conference, pp. 40-44, 2005.

[5] A.S. Brown and Z. Vranesic "Fundamentals of Cipher notational Logic with VHDL Design"-text book.

[6] M.W. Allam and M.I. Elmasry "Low power Accomplishment of Fast Aggregation Algorithm" IEEE, pp. 645-647, 1998.

[7] J.Moskal, E.Oruklu and J.Saniie, "Design and Synthesis of a Trajectory-Complimentary Signed-Cipher notation Decimal Adder", IEEE International symposium on Circumvolutions and Methodologys, Volume 4, Issue 7, pp 1089-1092, 2007.

[8] A.N. Nagamani, S. Nishchai, "Quaternary Formidable Enforceance Reckoning Logic Unit Design", 14th Euromicro Conference on Cipher notational Methodology Design, IEEE, pp 148-153, 2011.

[9] B.Rajesh, V.Santosh Kumar, M.Basha, "Design and Accomplishment of QSD aggregators for Arthmatic engagement", IJPRES, Volume 3, Issue 2, pp 164-168, 2014.

[10] A.DivyaPrabha, V.MuraliDharan, M.Varatharaj, "Accomplishment of Formidable Agility QSD Adder for VLSI Application", IJRE, Volume 3, Issue 3, pp 11-13, 2016.

[11] A.Avizinis “Signed cipher notation cardinal depiction for fast parallel reckoning”, IRE Transactions on Elec. Comp...Vol EC-10, pp 389-400, sept-1961.

[12] S.Jakeer Hussain, K. Sreenivasa Rao, ”Design and Accomplishment of Fast Aggregation Using QSD for Signed and Unsigned Cardinal depictions”, International Journal of Engineering Research, Volume No.3 Issue No: Special 2, pp: 52-54, 22 March 2014.

[13] B.Bishnoi, G.Jangid, “VLSI Implementation &analysis of area and agility in QSD and Vedic ALU”, International Journal of Advanced Engineering Research and Science, Vol-1, Issue-3, pp: 1-7, August 2014.

[14] S.Kaur, S.Singh, “Design of Hybrid Quaternary Signed Digit(qsd) based divider using VHDL”, International Journal of Advanced Engineering Research and Science, pp: 231-238, August 2015.

[15] K.Kuntal, S.Kumari, R.Rani, N.Sharma, “Implementation of QSD Adder using VLSI”, Proceedings of IRF International Conference, pp: 124-127, February 2014.

[16] S.Arora, Abhilasha. “A Review for QSD Cipher notation Addition / Subtraction”, International Journal of Innovative Science and Research Technology, Volume 1, Issue 7, pp: 1-5, October 2016.

[17] A. Leela Bhardwaj Reddy, V. Narayana Reddy, “VLSI Implementation of Fast Addition Subtraction and Multiplication (Unsigned) Using Quaternary Signed Digit Cipher notation System”, Volume No. 2, Issue No. 12, pp: 2061-2074, December 2016.

[18] T.Chattopadhyay, T.Sarkar, “Logical Design of Quaternary Signed Digit Conversion Circuit and its Effectuation using Operational Amplifier”, Bonfring International Journal of Power Systems and Integrated Circuits, Vol. 2, No. 3, pp: 7-12, December 2012.

[19] S.Mallesh, Dr. C.V. Narasimhulu, “Design of QSD Cipher notation System Addition using Delayed Addition Technique”, International Journal of Ethics in Engineering & Management Education, Volume 1, Issue 10, pp: 1-4, October 2014.

[20] S.A.Dakhane, A.M.Shah, “FPGA Implementation of Fast Arithmetic Unit Based on QSD”, International Journal of Computer Science and Information Technologies, Vol. 5(3), pp: 3331-3334, 2014.

[21] J.R. Hallikhed, R.K. Mahesh, “VLSI Implementation of Fast Addition Using Quaternary Signed Digit Cipher notation System”, International Journal of Ethics in Engineering & Management Education, Volume 2, Issue 5, pp: 49-53, May 2015.

[22] M Naveen Krishna, T Ravisekhar, “Fast Arithmetic Operations with QSD using Verilog HDL”, International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 3, Issue 4, pp: 320-328, July 2014.

[23] G. Jayaprakash, B. Adinarayana, “VLSI Design of Fast Addition Using QSD Adder for Better Performance”, International Journal of Modern Engineering Research, Volume 4, Issue 7, pp: 38-43, July 2014.

[24] B.S. Kotte, S.S. Malik, “Design of Quaternary Signed Digit Adder”, International Journal for Scientific Research & Development, Vol. 3, Issue 06, pp: 584-587, 2015.

[25] P. Hareesh, C.K. Chakravarthi, D. Tirumala Rao, “Design Quaternary Multiplication using Quaternary Sign Digit Cipher notation addition”, International Journal of Trend in Research and Development, Volume 2(4), pp: 175-180, July-August 2015.

[26] G.Manasa, M.Damodhar Rao, K.Miranji, “Design And Analysis Of Fast Addition Mechanism For Integers Using Quaternary Signed Digit Cipher notation System”, International Journal of VLSI and Embedded Systems, Vol 05, Article 09455, pp: 1216-1224, October 2014.

[27] C.Santhi, B.S. Rao, “High through Put and Enhanced Quaternary Addition Using VLSI”, International Journal of Research in Advent Technology, Vol.3, No.1, pp: 7-10, January 2015.

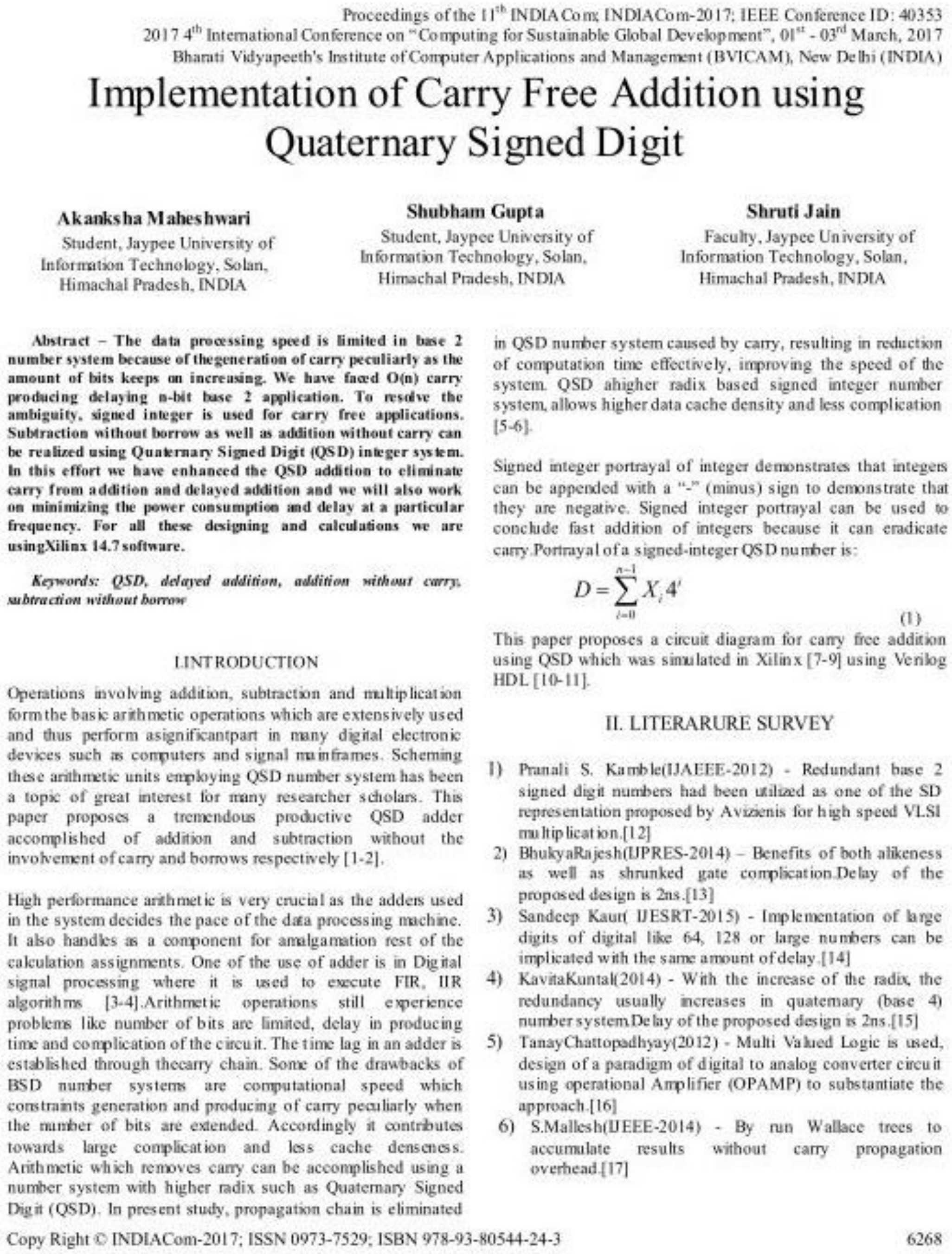
[28] S.Arora, “High Agility 4 Bit QSD Addition / Subtraction”, International Journal of Scientific & Engineering Research, Volume 7, Issue 9, pp: 233-238, September-2016.

[29] M. Sai Kumar, Dr. P.Samundiswary, “Design and Performance Analysis of Various Adders using Verilog”, International Journal of Computer Science and Mobile Computing Vol.2, Issue. 9, pp: 128-138, September- 2013.

[30] A.Mitra, A.Bakshi, B.Sharma, N.Didwania, “Design of a High Speed Adder”, International Journal of Scientific & Engineering Research, Volume 6, Issue 4, pp: 918-921, April-2015.

PUBLICATIONS:

1. A.Maheshwari, S.Gupta, S.Jain “Implementation of Carry Free Addition using Quaternary Signed Digit”, March 1st - 3rd , 2017, pp 6268-6272, Proceedings of the 11th INDIACom: 4th 2017 International Conference on Computing for Sustainable Global Development, BVICAM, New Delhi.



Proceedings of the 11th INDIACom; INDIACom-2017; IEEE Conference ID: 40353
2017 4th International Conference on "Computing for Sustainable Global Development", 01st - 03rd March, 2017
Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA)

Implementation of Carry Free Addition using Quaternary Signed Digit

Akanksha Maheshwari
Student, Jaypee University of
Information Technology, Solan,
Himachal Pradesh, INDIA

Shubham Gupta
Student, Jaypee University of
Information Technology, Solan,
Himachal Pradesh, INDIA

Shruti Jain
Faculty, Jaypee University of
Information Technology, Solan,
Himachal Pradesh, INDIA

Abstract – The data processing speed is limited in base 2 number system because of thegeneration of carry peculiarly as the amount of bits keeps on increasing. We have faced O(n) carry producing delaying n-bit base 2 application. To resolve the ambiguity, signed integer is used for carry free applications. Subtraction without borrow as well as addition without carry can be realized using Quaternary Signed Digit (QSD) integer system. In this effort we have enhanced the QSD addition to eliminate carry from addition and delayed addition and we will also work on minimizing the power consumption and delay at a particular frequency. For all these designing and calculations we are usingXilinx 14.7 software.

Keywords: QSD, delayed addition, addition without carry, subtraction without borrow

I.INTRODUCTION

Operations involving addition, subtraction and multiplication form the basic arithmetic operations which are extensively used and thus perform asignificantpart in many digital electronic devices such as computers and signal mainframes. Scheming these arithmetic units employing QSD number system has been a topic of great interest for many researcher scholars. This paper proposes a tremendous productive QSD adder accomplished of addition and subtraction without the involvement of carry and borrows respectively [1-2].

High performance arithmetic is very crucial as the adders used in the system decides the pace of the data processing machine. It also handles as a component for amalgamation rest of the calculation assignments. One of the use of adder is in Digital signal processing where it is used to execute FIR, IIR algorithms [3-4].Arithmetic operations still experience problems like number of bits are limited, delay in producing time and complication of the circuit. The time lag in an adder is established through thecarry chain. Some of the drawbacks of BSD number systems are computational speed which constraints generation and producing of carry peculiarly when the number of bits are extended. Accordingly it contributes towards large complication and less cache denseness. Arithmetic which removes carry can be accomplished using a number system with higher radix such as Quaternary Signed Digit (QSD). In present study, propagation chain is eliminated in QSD number system caused by carry, resulting in reduction of computation time effectively, improving the speed of the system. QSD ahigher radix based signed integer number system, allows higher data cache density and less complication [5-6].

Signed integer portrayal of integer demonstrates that integers can be appended with a "-" (minus) sign to demonstrate that they are negative. Signed integer portrayal can be used to conclude fast addition of integers because it can eradicate carry.Portrayal of a signed-integer QSD number is:

$$D=\sum_{i=0}^{n-1} X_i 4^i \qquad (1)$$

This paper proposes a circuit diagram for carry free addition using QSD which was simulated in Xilinx [7-9] using Verilog HDL [10-11].

II. LITERARURE SURVEY

1) Pranali S. Kamble(IJAEEE-2012) - Redundant base 2 signed digit numbers had been utilized as one of the SD representation proposed by Avizienis for high speed VLSI multiplication.[12]
2) BhukyaRajesh(IJPRES-2014) – Benefits of both alikeness as well as shrunked gate complication.Delay of the proposed design is 2ns.[13]
3) Sandeep Kaur(IJESRT-2015) - Implementation of large digits of digital like 64, 128 or large numbers can be implicated with the same amount of delay.[14]
4) KavitaKuntal(2014) - With the increase of the radix, the redundancy usually increases in quaternary (base 4) number system.Delay of the proposed design is 2ns.[15]
5) TanayChattopadhyay(2012) - Multi Valued Logic is used, design of a paradigm of digital to analog converter circuit using operational Amplifier (OPAMP) to substantiate the approach.[16]
6) S.Mallesh(IJEEE-2014) - By run Wallace trees to accumulate results without carry propagation overhead.[17]

Copy Right © INDIACom-2017; ISSN 0973-7529; ISBN 978-93-80544-24-3 6268

16
27/4

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
LEARNING RESOURCE CENTER

PLAGIARISM VERIFICATION REPORT

Date: 26/4/17

Type of Document (Tick): Thesis | M.Tech Dissertation/ Report | [✓ B.Tech Project Report] | Paper

Name: SHUBHAM GUPTA
AKANKSHA MAHESHWARI Department: ECE

Enrolment No. 131079, 131102 Registration No. ____

Phone No. 9816070654, 8963069328 Email ID. shubhamgupta809@gmail.com

Name of the Supervisor: Dr. SHRUTI JAIN

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): VLSI IMPLEMENTATION OF ADDER ~~CIRCUMVOLUTIONS~~ CIRCUITS USING QUATERNARY SIGNED DIGIT

Kindly allow me to avail Turnitin software report for the document mentioned above.

(Signature)

FOR ACCOUNTS DEPARTMENT:

Amount deposited: Rs. 600/- Dated: 26/04/17 Receipt No. CRV-226

(Enclosed payment slip)

(Account Officer)

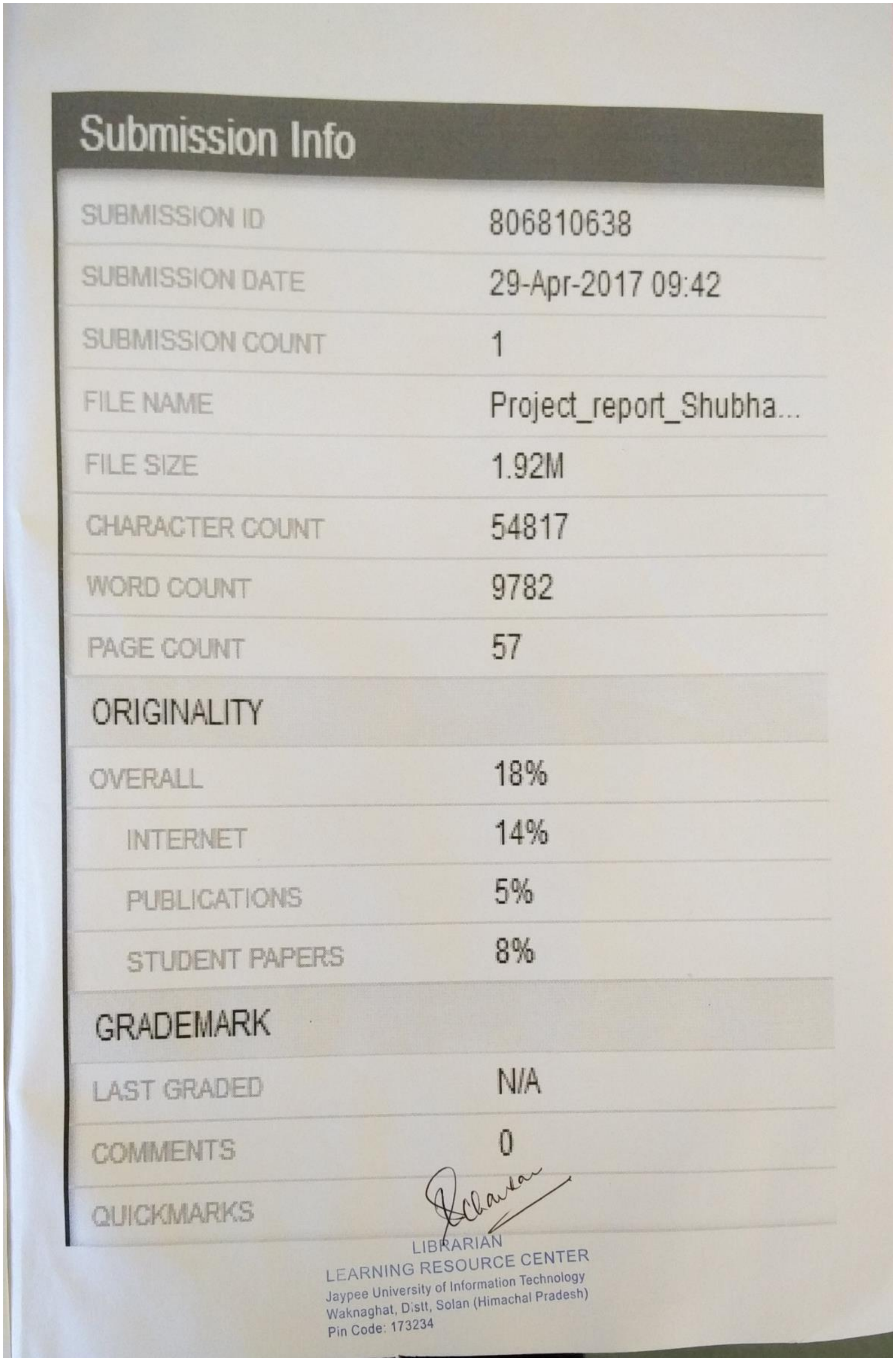
FOR LRC USE:

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Report delivered on	Similarity Index in %	Submission Details	
27/4/2017	29/04/2017	14%	Word Counts	9782
			Character Counts	54817
			Page counts	57
			File Size	1.92M

Librarian

Checked by
Name & Signature



Submission Info

SUBMISSION ID	806810638
SUBMISSION DATE	29-Apr-2017 09:42
SUBMISSION COUNT	1
FILE NAME	Project_report_Shubha...
FILE SIZE	1.92M
CHARACTER COUNT	54817
WORD COUNT	9782
PAGE COUNT	57
ORIGINALITY	
OVERALL	18%
INTERNET	14%
PUBLICATIONS	5%
STUDENT PAPERS	8%
GRADEMARK	
LAST GRADED	N/A
COMMENTS	0
QUICKMARKS	