Vol 05, Article 09455, October 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

DESIGN AND ANALYSIS OF FAST ADDITION MECHANISM FOR INTEGERS USING QUATERNARY SIGNED DIGIT NUMBER SYSTEM

G.MANASA[1], M.DAMODHAR RAO[2], K.MIRANJI[3]
[1]PG Student, ECE Department, Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India
[2]Assistant Professor, ECE Department, Gudlavalleru Engineering College, Gudlavalleru, AP, India
[3]Assistant Professor, ECE Department, Sir C R Reddy Engineering College, Eluru, AP, India
[1]maanasarao812@gmail.com, [2]damu406@gmail.com, [3]miranji.katta@gmail.com

ABSTRACT

With the binary number system, the speed of arithmetic operations are limited by formation and propagation of the carry. Using quaternary signed digit (QSD) number system both carry free addition and borrow free subtraction can be achieved. The QSD number system requires a special set of prime modulo based logic for arithmetic operations. Using a high radix number system such as Quaternary Signed Digit(QSD),a carry free arithmetic operation can be achieved. Arithmetic Operations such as addition and subtraction for large numbers like 64 and 128 can be computed without the propagation of carry using QSD number system. Design is simulated and analyzed using Xilinx 13.2 ISE Simulator.

Keywords-*Carry free addition, QSD*

1.INTRODUCTION

The speed of the digital processors depends mostly on the speed of the adders used in the system. Most of the arithmetic operations suffers from problems like limited number of bits, circuits complexity and delay. Carry look ahead adder produces a less propagation delay, but it is limited to small number of bits due to the circuit complexity. In this paper, a high speed QSD adder is proposed which is capable of performing carry free addition and borrow free subtraction using QSD numbers. For any operand size the QSD addition/subtraction operation employs a fixed number of minterms. In QSD number System carry propagation chains are eliminated which reduces the computation time.QSD is also advantages in case of parallelism and gate complexity.

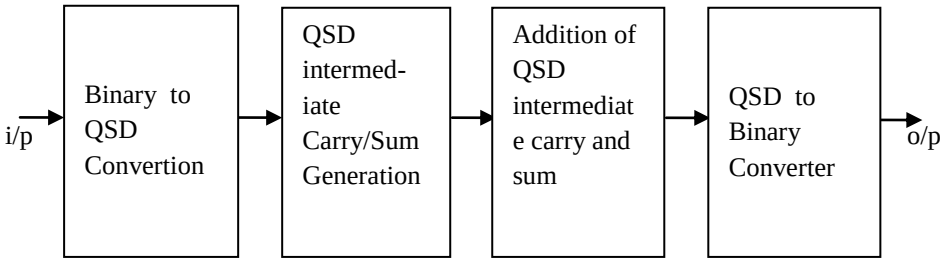


Fig:1 Block Diagram of QSD Addition

2. QUATERNARY SIGNED DIGIT(QSD) NUMBERS

QSD is a base-4 number system. Each signed-digit decimal number D can be represented in terms of an n digit quaternary signed digit number.

$$D= \sum_{i=0}^{n-1} x^i 4^i \quad \text{----------(1)}$$

Where x^i can be any value from the set $\{3^1,2^1,1^1, 0,1,2,3\}$.A QSD negative number is a QSD complement of the QSD positive number.

The advantage of QSD number system is that, it offers reduced circuit complexity both in terms of transistor count and interconnections. QSD number system consumes 25% less space when compared with the BSD number system and it can be verified with the help of the theorem given below:

To represent a numeric value $N\log_4 N$ number of QSD digits and $3\log_4 N$ binary bits are required while for the same $\log_2 N$ BSD digits and $2\log_2 N$ binary bits are required in BSD representation. Therefore QSD adder is better than RBSD adder as the QSD numbers saves $1/4^{th}$ of the storage space than the BSD numbers. So, the proposed QSD adder is beneficial in terms of number of gates, input connections and delay though both perform addition within constant time.

3.TECHNIQUE FOR CONVERSION FROM DECIMAL TO QUATERNARY NUMBER

1 digit QSD number can be represented using a 3-bit binary equivalent as

$3^1 = 101$

$2^1 = 110$

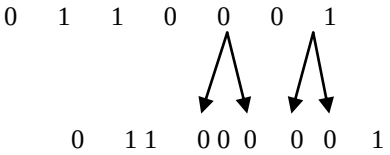
Vol 05, Article 09455, October 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

$1^{\bar{1}}$ = 111
0 = 000
1 = 001
2 = 010
3 = 011

So, to convert n-bit binary data into its equivalent q-digit QSD data, the n-bit binary data must be converted into3q-bit binary data. To achieve this we must split the 3rd,5th,7th bit…….i.e, odd bit(from LSB to MSB) into two portions. But we can't split the MSB bit. If the odd bit is 0 then, it is split into 0 & 0 and if it is 1 then, it is split into 1 & 0.

Example:

$(0110001)_2$ can be split as



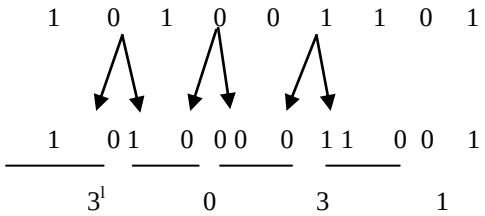
So we have to split the given binary data q-times. For example, the splitting is 1 time for conversion of a 2-digit quaternary number; the splitting is 2 times for conversion of a 3-digit quaternary number and so on. In each splitting one extra bit is generated. For conversion from binary to QSD, the required number of binary bits are

n=3q-{1*(q-1)} ----------(2)

So, for converting binary number into its equivalent QSD the number of bits should be 3,5,7,9 etc. According to the above equation every 3-bit can be converted into its QSD.

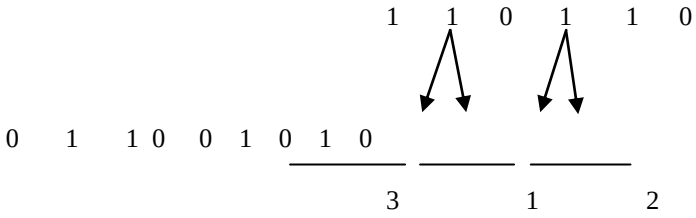
Examples for conversion from decimal to quaternary number:

1) Let $(-179)_{10}=(101001101)_2$ have to be converted into its equivalent QSD. It has 9 binary bits. Its 3rd bit is 1,5th bit is 0,7th bit is 1.So from equation(2),its equivalent QSD is of 4-digit.i.e,



So, the QSD equivalent of $(101001101)_2$ is $(3^{\bar{1}}031)_4$

2)Let$(54)_{10}$ =$(110110)_2$ have to be converted into its equivalent QSD. It has 16 binary bits. Its 3rd bit is 1,5th bit is 1.So from equation(2),its equivalent QSD is of 3-digit.i.e,



So, the QSD equivalent of $(110110)_2$ is $(312)_4$.

4.DESIGN PROCESS FOR QSD ADDER

In QSD number system the carry propagation chains are eliminated which in tern reduces the circuit complexity and interconnections, thus the speed of the machine increases. As the QSD number system ranges from -3 to +3,the results produced by the addition if two QSD numbers varies from -6 to +6.Table 1 represents the outputs for all possible combinations of two QSD numbers.

For the representation of decimal numbers in the range -3 to +3,one QSD digit is required. As the range of decimal number exceeds this range, more the one QSD digit is required. For the representation of addition result, which is in the range of -6 to +6,two QSD digits are required. Among the two digits in the QSD result,

Vol 05, Article 09455, October 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

the MSB digit represents the carry bit and the LSB digit represents the sum bit. In order to prevent this carry bit to propagate from lower digit position to higher digit position QSD number representation is used. In the number representation, QSD numbers allow redundancy. In QSD representation the same decimal number can be represented in more than one form. To prevent further rippling of carry one of the QSD number is chosen among the available QSD numbers.

The addition of two QSD numbers is done in tow steps to perform carry free addition. They are:

Step 1: From the given input QSD digits i.e,addend and augend, the first step generates an intermediate sum and intermediate carry.

Step 2: The intermediate sum of current digit is combined with the intermediate carry of the lower significant digit in step two.

So, the two QSD numbers are added in two stages.Stage-1 generates intermediate sum and intermediate carry for the given inputs.Stage-2 adds the intermediate carry of the lower significant digit with the current digits intermediate sum. There are two rules to perform QSD addition. These two rules helps to prevent the further rippling of carry.The Rules are:

Rule 1: The magnitude of the intermediate sum must be less than or equal to 2.Which means that it should be in the range of -2 to +2.

Rule 2: the magnitude of the intermediate carry must be less than or equal to 1.Which means that it should be in the range of -1 to +1.

Sum	QSD represented number	QSD ceded number
-6	$2^{1}2,1^{1}2^{1}$	$1^{1}2^{1}$
-5	$2^{1}3,1^{1}1^{1}$	$1^{1}1^{1}$
-4	$1^{1}0$	$1^{1}0$
-3	$1^{1}1,03^{1}$	$1^{1}1$
-2	$1^{1}2,02^{1}$	02^{1}
-1	$1^{1}3,01^{1}$	01^{1}
0	00	00
1	$01,13^{1}$	01
2	$02,12^{1}$	02
3	$03,11^{1}$	11^{1}
4	10	10
5	$11,23^{1}$	11
6	$12,22^{1}$	12

Table 1: Intermediate sum and carry between -6 to +6

By following the above mentioned rules the intermediate sum and carry from the step-1 have the range from -6 to +6.By utilizing the redundancy feature of QSD numbers we can choose the QSD represented numbers from the table-1 which satisfies the two rules mentioned above. The step-2 adds the intermediate sum of the current digit with the intermediate carry of the lower digit and produces the result which cannot be greater than 3 i.e, it will be in the range of -3 to +3.A single digit QSD number can be used to represent the result in this range i.e, -3 to +3.hence no further carry is required. The step-1 produces the output in the range of -6 to +6 which can be represented in intermediate sum and carry format as shown in the table-1.In the table-1,some numbers are having multiple QSD representations in column-2,but only those that meet the above mentioned two rules are chosen and are listed in the column-2 of the table-1 and are named as QSD coded numbers.

Example for QSD Addition:

To perform QSD addition of two numbers A=999 and B=349

Steps:

1)Convert the decimal number into its equivalent QSD representation

$A=(999)_{10}$

$=1*4^{5}+0*4^{4}+0*4^{3}+2^{1}*4^{2}+2*4^{1}+1^{1}*4^{0}$

Its QSD equivalent is $(1002^{1}21^{1})_{4}$

$B=(349)_{10}$

$=1*4^{4}+1*4^{3}+2*4^{2}+0*4^{1}+3^{1}*4^{0}$

Its QSD equivalent is $(11203^{1})_{4}$

2)Add the two QSD numbers

Vol 05, Article 09455, October 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

A=999		1	0	0	2^1	2	1^1
B=349				1	1	2	0 3^1
Decimal sum	1	1	1	0	2	-4	
IC	0	0	0	0	0	1^1	
IS		1	1	1	0	2	0
S		1	1	1	0	1	0
C_{out}	0						

The sum output is $(111010)_4$ which is equivalent to $(1348)_{10}$ and carry output is 0.

5.LOGIC DESIGN OF SINGLE DIGIT QSD ADDER

In carry free addition two steps are involved.Step-1 generates intermediate sum and carry from the given QSD inputs.Step-2 adds the intermediate carry of the lower digit with the intermediate sum of the current digit. Two rules are defined, to prevent further propagation of carry.Rule-1 defines that the intermediate sums magnitude must be less than or equal to 2.Rule-2 defines that the intermediate carry's magnitude must be less than or equal to 1.By this the magnitude of the step-2 output cannot be greater than 3 which can be represented using a single digit QSD number. All possible pairs of inputs to step-1 are considered and the range varies from -3 to +3.So the result of addition varies from -6 to +6 which needs two QSD digits. The propagation of carry can be eliminated by mapping the two digits into a pair of intermediate sum and carry such that the nth intermediate sum and (n-1)th intermediate carry never form any carry generating pair (3,3),(3,2),(3,1),(3^1,3^1),(3^1,2^1),(3^1,1^1).The carry free addition can be achieved by restricting the representation such that all the intermediate sums are restricted to be less than or equal to 2 and all the intermediate carry's are restricted to be less than or equal to 1.The mapping between the inputs, outputs, intermediate sum and carry are shown in table-2.

The redundancy feature of QSD numbers is used to prevent further propagation of carry. The range of intermediate carry is in between -1 & +1,so it can be represented using 2-bit binary number. But for compatibility with the intermediate sum 3-bit binary representation is taken. A 3 variable input A_2,A_1,A_0 is used to represent the addend A_i and a 3 variable input B_2,B_1,B_0 is used to represent the augend B_i at the input side.IS_2,IS_1,IS_0 are used to represent the intermediate sum and IC_2,IC_1,IC_0 are used to represent the intermediate carry at the output side. The six variable expressions for intermediate sum and carry in terms of inputs can be derived from the table-2.So 6 output expressions are obtained for IS_2,IS_1,IS_0,IC_2,IC_1 and IC_0.Here the third appended bit IC_2 is equal to IC_1 because the intermediate carry can be represented by only 2-bits.So the expression for both outputs will be same. For the generation of intermediate sum and carry the logic equations specifying the minimal hardware realization are derived by using a 6-variable K-map.

The minterms for the intermediate sum are:

$IS0=a_0b_0{}^1+a_0{}^1b_0$

$IS1=(a_1b_1{}^1+a_1{}^1b_1)(a_0b_0)^1+ (a_1b_1{}^1+a_1{}^1b_1)^1(a_0b_0)$

$IS2=IS0(a_1{}^1b_1+a_1b_1{}^1)+b_2(a_1b_0)^1+ a_1(b_1a_0)^1+a_0b_0(a_1b_1)^1$

The minterms for intermediate carry are:

$IC_2=IC_1=(a_2b_2)(a_0b_0a_1b_1)^1+(a_1+ b_1)^1(a_2b_0{}^1+a_0{}^1b_2)$

$IC_0=IC_2+(a_2b_2)^1(a_1b_1+b_1b_0+a_1b_0+a_0b_1+ a_1a_0)$

The final sum which is carry free is generated from the above outputs. Therefore it has 6 inputs and 3 outputs bits.

$S_0=IC_0IS_0{}^1+ IC_0{}^1IS_0$

$S_1=IC_1 \oplus IS_1 \oplus IC_0IS_0$

$S_2= IC_2 \oplus IS_2 \oplus (IC_1IS_1+(IC_1+ IS_1)IC_0IS_0)$

In QSD addition, parallel addition is done i.e, the addition of higher order digits does not wait until the addition of lower order digits is completed.

Vol 05, Article 09455, October 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

Input			Output			
QSD	Binary		Decimal	QSD		Binary
Ai Bi	Bi	Ai	Sum Si	Ci	Si	Ci
3 011	-3	011	6 010	1	2	001
3 010	2	011	5 001	1	1	001
2 011	3	010	1 001	1	1	001
3 001	1	011	4 000	1	0	001
1 011	3	001	4 000	1	0	001
2 010	2	010	4 000	1	0	001
1 010	2	001	3 111	1	0	001
2 001	1	010	3 111	1	0	001
3 000	0	011	3 111	1	0	001
0 011	3	000	3 111	1	0	001
1 001	1	001	2 010	0	2	000
0 2	000	010	2 010	0	2	000
2 000	0	010	2 010	0	2	000
3 111	-1	011	2 010	0	2	000
-1 011	3	111	2 010	0	2	000
0 001	1	000	1 001	0	1	000
1 000	0	001	1 001	0	1	000
2 111	-1	010	1 001	0	1	000
-1 010	2	111	1 001	0	1	000
3 110	-2	011	1 001	0	1	000
-2 011	3	110	1 001	0	1	000
0 000	0	000	0 000	0	0	000
1 111	-1	001	0 000	0	0	000
-1 001	1	111	0 000	0	0	000
2 110	-2	010	0 000	0	0	000
-2 010	2	110	0 000	0	0	000
-3 011	3	101	0 000	0	0	000

Vol 05, Article 09455, October 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

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

Table 2:The intermediate sum and carry for all combinations of inputs

Vol 05, Article 09455, October 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

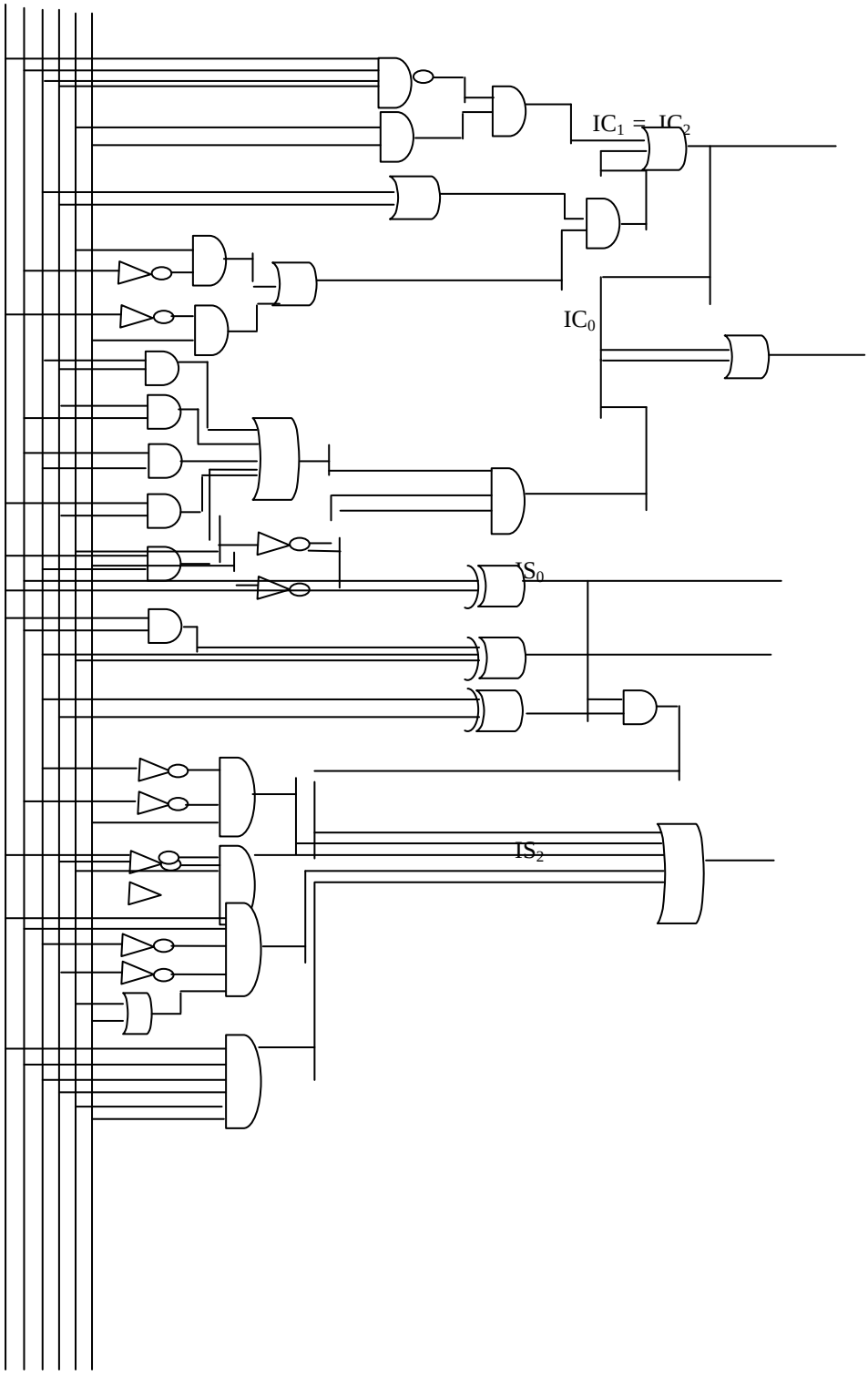


Fig 2:Date Flow of Single Digit QSD Adder Cell

6.SIMULATION RESULTS

The code for QSD adder is written in verilog and the design is simulated and analyzed using Xilinx 13.2 ISE Simulator.

Vol 05, Article 09455, October 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

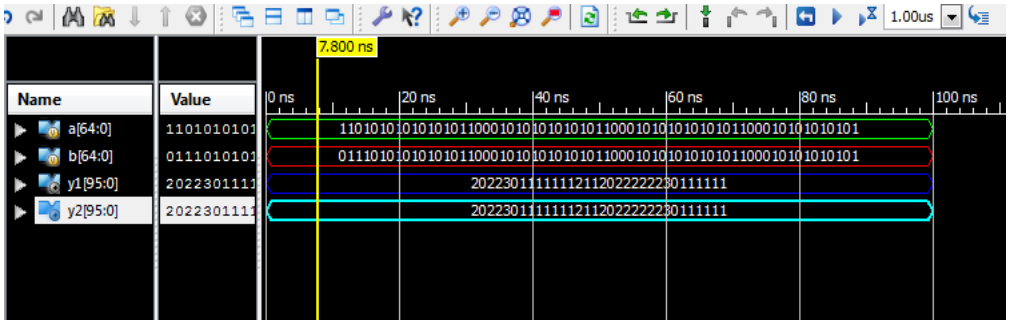


Fig:3 Simulation Result of conversion from binary to QSD

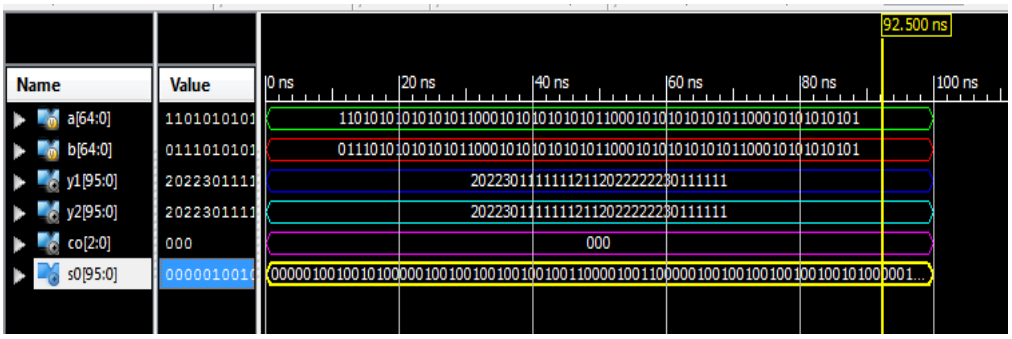


Fig:4 Simulation Result of QSD Addition

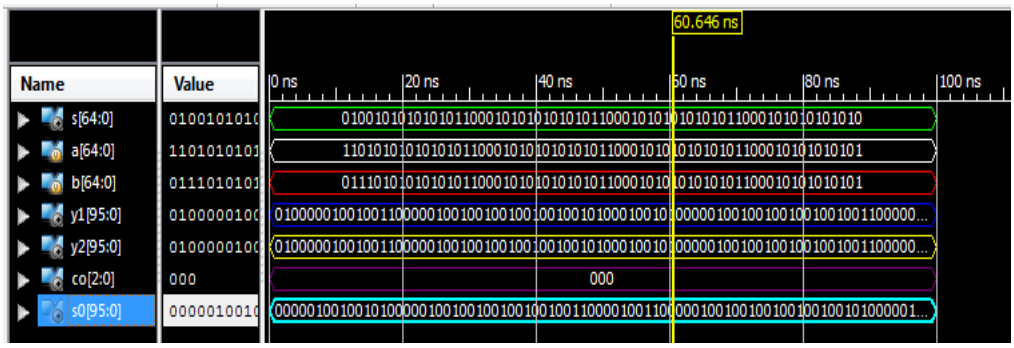


Fig:5 Simulation Result of QSD Addition and decimal output

7.ANALYSIS REPORT

Components	Analysis Report
IO's	261
4-Input LUT's	256
CLB's	147
Delay	80.137ns
XOR Gates	194

Table:3 Analysis Report for 65-bit CLA Adder

Components	Analysis Report
IO's	196
4-Input LUT's	129
CLB's	74
Delay	72.616ns
XOR Gates	65

Table:4 Analysis Report for 64-bit QSD Adder

Vol 05, Article 09455, October 2014 International Journal of VLSI and Embedded Systems-IJVES
http://ijves.com ISSN: 2249 – 6556

7.CONCLUSION

In this paper, the technique for conversion from binary number system to quaternary number system is explained with examples. For the addition of the Quaternary number a different technique is used i.e, for the addition of two QSD numbers first the intermediate sum and carry are generated and then these intermediate sum and carry are added with each other with the help of full adders. The result of this addition is always carry free. Here since, for the conversion from binary to QSD, the splitting of bits is done for LSB to MSB for every odd bit. The splitting should stop at the odd bit and the MSB bit cannot be split out because it is a sign bit. So, this paper implements the QSD addition for a 65-bit number. This process of developing a quaternary number system and addition of QSD numbers using quaternary addition consumes a delay of 72.616ns,74 CLB's,129 4-input LUT's,196 IO's and 65 XOR gates.. The adder does not produce any carry which in turn reduces the complexity and interconnections. The addition using QSD number system is compared with addition using binary number system and the results are analyzed.

ACKNOWLEDGEMENT

I am glad to express my deep sense of gratitude to, **Mr. M.DAMODHAR RAO,** Assistant Professor of Electronics and Communication Engineering, for his guidance and cooperation in completing this project. Through this, I want to convey my sincere thanks to him for his inspiring assistance during my project. I express my heartfelt gratitude and deep indebtedness to the beloved Head of the Department, **Dr. M.KAMARAJU,** for his great help and encouragement in completion of my project. I also express my gratitude to our principal **Dr. P.NAGESWARA REDDY,** for his encouragement and facilities provided for my project. I thank one and all who have rendered help to us directly or indirectly in the completion of this work.

REFERENCES

[1] Reena Rani, Neelam Sharma, L.K.Singh, "FPGA Implementation of Fast Adders using Quaternary Signed Digit Number System" IEEE proceedings of International Conference on Emerging Trends in Electronic and Photonic Devices & Systems (ELECTRO-2009), pp 132-135, 2009.
[2] Reena Rani, Neelam Sharma, L.K.Singh, "Fast Computing using Signed Digit Number System" IEEE proceedings of International Conference on Control, Automation, Communication And Energy Conservation - 2009, 4th-6th June 2009.
[3] Reena Rani, L.K. Singh and Neelam Sharma, "A Novel design of High Speed Adders Using Quaternary Signed Digit Number System "International Journal of Computer and Network Security(IJCNS), Vol. 2,No. 9, pp.62-66, September 2010.
[4] Vasundara Patel K S, K S Gurumurthy, "Design of High Performance Quaternary Adders", International Journal of Computer Theory and Engineering, Vol.2, No.6, pp. 944-952, December, 2010.
[5] A.A.S. Awwal and J.U. Ahmed, "fast carry free adder design using QSD number system ,"proceedings of the IEEE 1993 national aerospace and electronic conference, vol 2,pp 1085-1090,1993.
[6] A.A.S Awwal, Syed M. Munir, A.T.M. Shafiqul Khalid, Howard E. Michel and O. N. Garcia, "Multivalued Optical Parallel Computation Using An Optical Programmable Logic Array", Informatica, vol. 24, No. 4, pp. 467-473, 2000.
[7] Behrooz Parhami, "Carry-Free Addition of Recoded Binary Signed- Digit Numbers", IEEE Transactions on Computers, Vol. 37, No. 11, pp. 1470-1476, November 1988.
[8] S. Hurst, "Multiple-valued logic -its status and its future", IEEE trans.On Computers. Vol. C-33, no.12, pp. 1160-1179, 1984.
[9] Hanyu, M. Kameyama, "A 200 MHz pipelined multiplier using 1.5V-supply multiple valued MOS current-mode circuits with dual-rail source-coupled logic", IEEE Journal of Solid-State Circuits vol.30, no.11, pp.1239-1245, 1995.
[10] Kawahito, S. Kameyama, "A 32 X 32 bit Multiplier using Multiple-valued MOS Current Mode Circuit", Journal of Solid-State Circuits, IEEE, vol.1, pp.124 - 132, 1988.
[11] Y. Yasuda, Y. Tokuda, S. Zhaima, K. Pak, T.Nakamura,A.Yoshida, "Realization of quaternary logic circuits by N-Channel MOS Devices", IEEE Journal of Solid State Circuits, vol.21, no.1,pp.162-168, 1986.
[12] S. L. Hurst, "Multiple-Valued Logic—Its Status and its Future". IEEE Transactions computers, Vol. C-33, No. 12, Pp. 1160-1179, 1984.
[13] T. Chattopadhyay, G.K. Maity and Jitendra Nath Roy, "Designing of all-optical tri-state logic system with the help of optical nonlinear material", Journal of Nonlinear Optical Physics & Materials, Vol. 17, No. 3,Pp.315-328, 2008.
[14] W. Clark, J. Lian, "On arithmetic weight for general radix representation of integers". IEEE Trans. Inform. Theory, Vol.19, Pp.823-826, 1973.
[15] T. Chattopadhyay, J.N. Roy, "Easy conversion technique of binary to quaternary signed digit and vice versa." hysics Express, Vol. 1, No. 3, Pp. 165-174, 2011.