 ISSN: 2321-8169
280 – 285

Design of QSD Multiplier Using VHDL

Pooja s. Rade[1], Ashwini M. Khode[2], Rajani N. Kapse[3],
Ankita M. Sor[4], Smruti G. Bhasme[5],Prashant Y. Shende[6]
[1-5] (BE 4th Year Student) and [6](Assistant profesor)
[1-6] (Department of ENTC, DMIETR Wardha, Maharashtra, India.)

Abstract**—**The need for high speed digital circuits became more prominent as portable multimedia and communication applications incorporating information processing and computing. The drawback of modern computers lead to the worsening in performance of arithmetic operations such as addition, subtraction, multiplication on the aspects of carry propagation time delay, high power consumption and large circuit complexity.Binary Signed Digit Numbers are known to allow limited carry propagation with more complex addition process. Some of the limitations of this system are computational speed which limits formation and propagation of carry especially as the number of bits increases. Therefore it provides large complexity and low storage density. Carry free arithmetic operations can be achieved using a higher radix number system such as Quaternary Signed Digit (QSD) and it allows higher information storage density, less complexity. A high speed area effective adders and multipliers can be implemented using this technique. Carry free addition and other operations on a large number of digits such as 64, 128, or more can be implemented with constant delay and less complexity. The Design is simulated & synthesized using Xilinx 13.1.

Keywords*-VHDL; QSD (Quaternary Signed Digit); Programmable Logic; Fast Computation*

*****

I. INTRODUCTION

Multipliers are one of the most important arithmetic units in microprocessors and DSPs and also a major source of power dissipation. Reducing the power dissipation of multipliers is key to satisfying the overall power budget of various digital circuits and systems.Modern computers are based on binary number system (radix =2). They have two logical states ‘0’ and ‘1’.In such system, ‘1’ plus ‘1’ is ’0’ with carry ‘1’ (i.e. 1+1=10). This carry should have to add with another ‘1’, as a result further carry ‘1’ generates. This creates the delay problem in computer circuits [1]. So to get rid of this carry formation again and again signed digit is essential. For high-speed arithmetical calculation, carry free adders improves the operational performance. Arithmetic operations are widely used and play important roles in various digital systems such as computers and signal processors [2]. Many researchershave attracted by the QSD number representation. As the arithmetic operations still suffer from known problems including limited number of bits, propagation time delay, and circuit complexity. In present study, QSD number system eliminates carry propagation chain which reduces the computation time substantially, thus enhancing the speed of the machine. QSD Adder or QSD Multiplier circuits are logic circuits designed to perform high-speed arithmetic operations [3].
This paper proposes a high speed QSD multiplication operation by using 3 Bit QSD adder.The QSD addition operation employs a fixed number of minterms for any operand size. The multiplier is composed of partial product generators and adders. For convenience oftesting and to verify results.

This paper is organized as follows. Section II presents the quaternary signed digit (QSD) number system. Section III contains basic concept of the project. Conversion technique given in section IV. The QSD multiplier design together with QSD adder along with simulation results detailed in sectionV and section VI respectively. Section VII presents conclusions and future scope. References are shown in section VIII.

II. QSD NUMBER SYSTEM

Quaternary is the base 4-numeral system. It uses the digits 0, 1, 2 and 3 to represent any real number. It shares with all fixed-radix numeral systems. It has the ability to represent any real number with a canonical representation (almost unique) and the characteristics of the representations of rational numbers and irrational numbers. See decimal and binary for a discussion of these properties.

Relation to binary

Quaternary has a special relation to the binary numeral system. Each radix 4, 8 and 16 is a power of 2, so the conversion to and from binary is implemented by matching each digit with 2, 3 or 4 binary digits, or bits [5].
QSD numbers are represented using 3-bit 2’s complement notation. To produce anappropriate decimal representation, each number can be represented by

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

Where, x_i can be any value from the set $\{\bar{3},\bar{2},\bar{1},0,1,2,3\}$
A QSD negative number is the QSD complement of the QSD positive number i.e., $\bar{3}= -3$,$2= -2$ and $\bar{1}= -1$.
For example,
$1\overline{233}_{QSD}= 1*4^3+\bar{2}*4^2+\bar{3}*4^1+\bar{3}*4^3$
$= 23_{10}$
and$\bar{1}23\bar{3}_{QSD}= -23_{10}$

Comparison of QSD with BSD

It offers the advantage of reduced circuit complexity, i.e. number of transistor required is less and minimum interconnections are needed. According to this theorem QSD number uses 25% less space than BSD to store number [6]. Theorem is described as under- to represent numeric value N, $\log_4 N$ number of QSD digits and 3 $\log_4 N$ binary bits are required. And for BSD representation of same number $\log_2 N$

 ISSN: 2321-8169
280 – 285

BSD digits and $2\log_2 N$ binary bits are required. The ratio of number of bits required in QSD representation to the number of bits required in BSD representation for an any number N is

$$\frac{|3\log_4 N|}{|2\log_4 N|} = \frac{3\frac{\log N}{\log 4}}{2\frac{\log N}{\log 2}} = \frac{3\log 2}{2\log 4} = \frac{3}{4}$$

Therefore, QSD saves ¼ of the storage used by BSD.Also it Reduce the computation time. In general the number of bits required by a QSD number system is less when compared to BSD number system, which in turn results in better speeds and performance.

III. Basic Concept

The general block diagram of QSD multiplier is shown in figure 1 below.

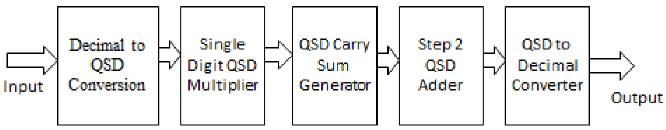


Figure.1 General Block Diagram of QSD Multiplier

To perform any operation in QSD, first convert the binary or any other input into quaternary signed digit.

IV. Decimal to QSD Conversion

Single digit QSD number can be represented by using a 3-bit binary equivalent are 3 =011

2 =010

1 =001

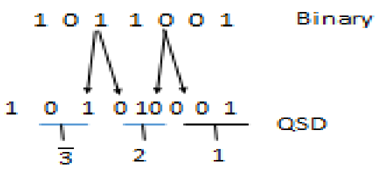
0 = 000

$\bar{3}$= 101

$\bar{2}$= 110

$\bar{1}$= 111

To convert n-bit binary data to its equivalent m-digit QSD data, we have to convert this n-bit binary data into 3 m-bit binary data to achieve the target we have to split odd bit from LSB to MSB i.e. 3,5,7 bit into two portions. But we cannot split the MSB. If the odd bit is one then, it is split into 1 and 0 and if it is 0 then, it is split into 0 and 0, the splitting technique of binary number 10011001



Rules for carry free operation

To remove the further rippling of carry there are two rules to perform QSD multiplication in two steps:

Rule 1: First rule states that the magnitude of the intermediate sum must be less than or equal to 2 i.e., it should be in the range of -2 to +2.

Rule 2: Second rule states that the magnitude of the intermediate carry must be less than or equal to 1 i.e., it should be in the range of -1 to +1 [4].

By considering the Rules for carry free multiplication. Some numbers have multiple representations, but only those that meet the defined rules are chosen. The chosen intermediate carry andsum and product are listed in the last column of Table:1

Table1: QSD Number Representation for Carry free Addition and Multiplication

Sum and product	QSD represented number	QSD coded number
-9	$\overline{21}$, $\overline{3}3$	$\overline{21}$
-6	$\bar{1}2$, $\bar{2}2$	$\bar{1}\,\bar{2}$
-5	$\overline{23}$, $\overline{11}$	$\overline{11}$
-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	00	0 0
1	01 , $0\bar{3}$	0 1
2	02, $1\,\bar{2}$	0 2
3	03, $1\,\bar{1}$	$1\,\bar{1}$
4	10	1 0
5	11, $2\bar{3}$	11
6	12 , $2\bar{2}$	12
9	21, $3\bar{3}$	21

Steps for Carry free operation

To perform carry free operation,The multiplication of two whole numbers is equivalent to the addition of one of them with itself. The addition of two QSD numbers can be done in two steps [4]: Thus for multiplier it will required 3 steps.

Step1: First generates an product Mi and carry C_i which is applied to intermediate carry sum generator i.e. 1st step of QSD adder.

Step 2: second step generates an intermediate carry and intermediate sum from the input QSD digits.

Step 3: Third step combines intermediate sum of current digit with the intermediate carry of the lower significant digit.

The implementation of an n-digit partial product generator uses n units of the single-digit QSD multiplier. Shown in block diagram figure 2.

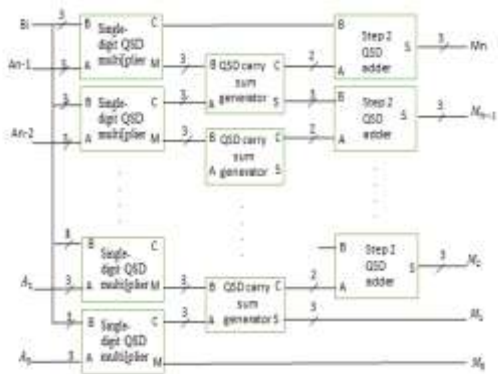


Figure 2: n-Digit QSD Multiplier

 ISSN: 2321-8169
280 – 285

V. SINGLE DIGIT QSD MULTIPLIER

There are generally two methods for a multiplication operation: parallel and iterative.The multiplication of two whole numbers is equivalent to the addition of one of them with itself as many times as the value of the other one. All possible input pairs of the addend and augend are considered. The output ranges from -9 to 9 as shown in Table 2.

Table 2: The outputs of all possible combinations of a pair of multiplicand (A) and multiplier (B).

B \ A	-3	-2	-1	0	1	2	3
-3	9	6	3	0	-3	-6	-9
-2	6	4	2	0	-2	-4	-6
-1	3	2	1	0	-1	-2	-3
0	0	0	0	0	0	0	0
1	-3	-2	-1	0	1	2	3
2	-6	-4	-2	0	2	4	6
3	-9	-6	-3	0	3	6	9

The QSD representation of a single digit multiplication output, M as a result and C as a carry to be combined with M of the next digit. The range of both outputs, M and C, isbetween -2 and 2.shown in Table 3, contains a carry-out of magnitude 2 when the output is either -9 or 9.

Table 3: The mapping between the inputs and outputs of the multiplier

INPUT				OUTPUT				
QSD		Binary		Decimal	QSD		Binary	
Ai	Bi	Ai	Bi	Product	C_i	Mi	Ci	Mi
1	$\bar{3}$	001	101	-3	$\bar{1}$	1	111	001
1	$\bar{2}$	001	110	-2	0	$\bar{2}$	000	110
1	$\bar{1}$	001	111	-1	0	$\bar{1}$	000	111
1	0	001	000	0	0	0	000	000
1	1	001	001	1	0	1	000	001
1	2	001	010	2	0	2	000	010
1	3	001	011	3	1	$\bar{2}$	001	111
$\bar{2}$	$\bar{3}$	110	101	6	1	2	001	010
$\bar{2}$	$\bar{2}$	110	110	4	1	0	001	000
$\bar{2}$	$\bar{1}$	110	111	2	0	2	000	010
$\bar{2}$	0	110	000	0	0	0	000	000
$\bar{2}$	1	110	001	-2	0	$\bar{2}$	000	110
$\bar{2}$	2	110	010	-4	$\bar{1}$	0	111	000
$\bar{2}$	3	110	011	-6	$\bar{1}$	$\bar{2}$	111	110
$\bar{1}$	$\bar{3}$	111	101	3	1	$\bar{1}$	001	111
$\bar{1}$	$\bar{2}$	111	110	2	0	2	000	010
$\bar{1}$	$\bar{1}$	111	111	1	0	1	000	001
$\bar{1}$	0	111	000	0	0	0	000	000
$\bar{1}$	1	111	001	-1	0	$\bar{1}$	000	111
$\bar{1}$	2	111	010	-2	0	$\bar{2}$	000	110
$\bar{1}$	3	111	011	-3	$\bar{1}$	1	111	001
0	$\bar{3}$	000	101	0	0	0	000	000
0	$\bar{2}$	000	110	0	0	0	000	000
0	$\bar{1}$	000	111	0	0	0	000	000
0	0	000	000	0	0	0	000	000
0	1	000	001	0	0	0	000	000
0	2	000	010	0	0	0	000	000
0	3	000	011	0	0	0	000	000
$\bar{3}$	$\bar{3}$	101	101	9	2	1	010	001
$\bar{3}$	$\bar{2}$	101	110	6	$\bar{1}$	$\bar{2}$	111	110
$\bar{3}$	$\bar{1}$	101	111	3	1	$\bar{1}$	001	111
$\bar{3}$	0	101	000	0	0	0	000	000
$\bar{3}$	1	101	001	-3	$\bar{1}$	1	111	001
$\bar{3}$	2	101	010	-6	$\bar{1}$	$\bar{2}$	111	110
$\bar{3}$	3	101	011	-9	$\bar{2}$	$\bar{1}$	110	111
2	$\bar{3}$	010	101	-6	$\bar{1}$	$\bar{2}$	111	001
2	$\bar{2}$	010	110	-4	$\bar{1}$	0	111	000
2	$\bar{1}$	010	111	-2	0	$\bar{2}$	000	110
2	0	010	000	0	0	0	000	000
2	1	010	001	2	0	2	000	010
2	2	010	010	4	1	0	001	000
2	3	010	011	6	1	2	001	010
3	$\bar{3}$	011	101	-9	$\bar{2}$	$\bar{1}$	110	111
3	$\bar{2}$	011	110	-6	$\bar{1}$	$\bar{2}$	111	110
3	$\bar{1}$	011	111	-3	$\bar{1}$	1	111	001
3	0	011	000	0	0	0	000	000
3	1	011	001	3	1	$\bar{1}$	001	111
3	2	011	010	6	1	2	001	010
3	3	011	011	9	2	1	010	001

The single-digit multiplication produces According to Table 3,the diagram of a single-digit QSD multiplier is shown in Figure 3.

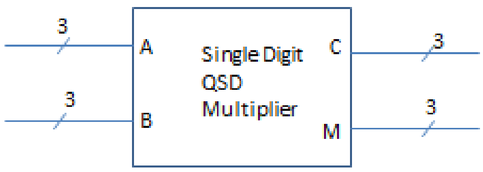


Figure 3: QSD single Digit Multiplier

QSD multiplication can be implemented in both ways, requiring a QSD partial product generator and QSD adder as abasic components.
In the 2^{nd} step of multiplier i.e.1^{st} step of QSD adder, the range of the output is from -6 to 6 which can be represented in the intermediate carry and sum in QSD format as show in Table 2.

Table 2:The outputs of all possible combinations of a pair of intermediate carry (A) and sum (B).

ISSN: 2321-8169
280 – 285

B \ A	-3	-2	-1	0	1	2	3
-3	-6	-5	-4	-3	-2	-1	0
-2	-5	-4	-3	-2	-1	0	1
-1	-4	-3	-2	-1	0	1	2
0	-3	-2	-1	0	1	2	3
1	-2	-1	0	1	2	3	4
2	-1	0	1	2	3	4	5
3	0	-1	2	3	4	5	6

Both inputs and outputs can be encoded in 3-bit 2's complement binary number. The mapping between the inputs, addend and augend, and the outputs, the intermediate carry and sum are shown in binary format in Table 3.

Table 3:Mapping Between input and outputs of intermediate carry and sum

INPUT				OUTPUT				
QSD		Binary		Decimal	QSD		Binary	
Ai	Bi	Ai	Bi	Sum	C_i	S_i	C_i	S_i
3	3	011	011	6	1	2	01	010
3	2	011	010	5	1	1	01	001
2	3	010	011	5	1	1	01	001
3	1	011	001	4	1	0	01	000
1	3	001	011	4	1	0	01	000
2	2	010	010	4	1	0	01	000
1	2	001	010	3	1	$\bar{1}$	01	111
2	1	010	001	3	1	$\bar{1}$	01	111
3	0	011	000	3	1	$\bar{1}$	01	111
0	3	000	011	3	1	$\bar{1}$	01	111
1	1	001	001	2	0	2	00	010
0	2	000	010	2	0	2	00	010
2	0	010	000	2	0	2	00	010
3	$\bar{1}$	011	111	2	0	2	00	010
$\bar{1}$	3	111	011	2	0	2	00	010
0	1	000	001	1	0	1	00	001
1	0	001	000	1	0	1	00	001
2	$\bar{1}$	010	111	1	0	1	00	001
$\bar{1}$	2	111	010	1	0	1	00	001
3	$\bar{2}$	011	110	1	0	1	00	001
$\bar{2}$	3	110	011	1	0	1	00	001
0	0	000	000	0	0	0	00	000
1	$\bar{1}$	001	111	0	0	0	00	000
$\bar{1}$	1	111	001	0	0	0	00	000
2	$\bar{2}$	010	110	0	0	0	00	000
$\bar{2}$	2	110	010	0	0	0	00	000
$\bar{3}$	3	101	011	0	0	0	00	000
3	$\bar{3}$	011	101	0	0	0	00	000
0	$\bar{1}$	000	111	-1	0	$\bar{1}$	00	111
$\bar{1}$	0	111	000	-1	0	$\bar{1}$	00	111
$\bar{2}$	1	110	001	-1	0	$\bar{1}$	00	111
1	$\bar{2}$	001	110	-1	0	$\bar{1}$	00	111
$\bar{3}$	2	101	010	-1	0	$\bar{1}$	00	111
2	$\bar{3}$	010	101	-1	0	$\bar{1}$	00	111
$\bar{1}$	$\bar{1}$	111	111	-2	0	$\bar{2}$	00	110
0	$\bar{2}$	000	110	-2	0	$\bar{2}$	00	110
$\bar{2}$	0	110	000	-2	0	$\bar{2}$	00	110
$\bar{3}$	1	101	001	-2	0	$\bar{2}$	00	110
1	$\bar{3}$	001	101	-2	0	$\bar{2}$	00	110
$\bar{1}$	$\bar{2}$	111	110	-3	$\bar{1}$	1	11	001
$\bar{2}$	$\bar{1}$	110	111	-3	$\bar{1}$	1	11	001
$\bar{3}$	0	101	000	-3	$\bar{1}$	1	11	001
0	$\bar{3}$	000	101	-3	$\bar{1}$	1	11	001
$\bar{3}$	$\bar{1}$	101	111	-4	$\bar{1}$	0	11	000
$\bar{2}$	$\bar{3}$	111	101	-4	$\bar{1}$	0	11	000
$\bar{1}$	$\bar{2}$	110	110	-4	$\bar{1}$	0	11	000
$\bar{3}$	$\bar{2}$	101	110	-5	$\bar{1}$	$\bar{1}$	11	111
$\bar{2}$	$\bar{3}$	110	101	-5	$\bar{1}$	$\bar{1}$	11	111
$\bar{3}$	$\bar{3}$	101	101	-6	$\bar{1}$	$\bar{2}$	11	110

Table 3 shows all possible combinations of the summation between the intermediate carry and the sum.

Since the intermediate carry is always between -1 and 1, it requires only a 2-bit binary representation. The intermediate carry and sum circuit is shown in Figure4.

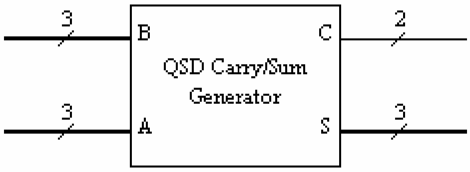


Figure 4. The intermediate carry and sum generator

In step 3, the intermediate carry from the lower significant digit is added to the sum of the current digit to produce the final result. The addition in this step produces no carry because the current digit can always absorb the carry-in from the lower digit.

Table 4:The outputs of all possible combinations of a pair of intermediate carry (A) and sum (B).

A \ B	-2	-1	0	1	2
-1	-3	-2	-1	0	1
0	-2	-1	0	1	2
1	-1	0	1	2	3

Theresultof addition in this steprangesfrom -3 to 3. Since carry is not allowed in this step, the result becomes a single digit QSD output. The output is a 3-bit binary represented QSD number. Figure 5 shows the diagram of the second step adder.

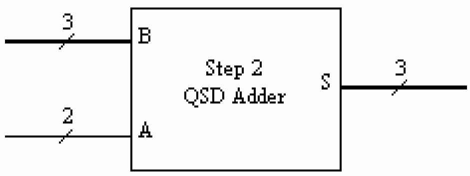


Figure 5. The second step QSD adder.

The mapping between the 5- bit input and the 3-bit output is shown in Table 5.

Table 5: The mapping between the inputs and outputs of the intermediate carry and sum.

INPUT				OUTPUT		
QSD		Binary		Decimal	QSD	Binary
A_i	B_i	A_i	B_i	Sum	S_i	S_i
1	2	01	010	3	3	111
1	1	01	001	2	2	010
0	2	00	010	2	2	010
0	1	00	001	1	1	001
1	0	01	000	1	1	001
$\bar{1}$	2	11	010	1	1	001
0	0	00	000	0	0	000
1	$\bar{1}$	01	111	0	0	000
$\bar{1}$	1	11	001	0	0	000
0	$\bar{1}$	00	111	-1	$\bar{1}$	111
$\bar{1}$	0	11	000	-1	$\bar{1}$	111
1	$\bar{2}$	01	110	-1	$\bar{1}$	111
$\bar{1}$	$\bar{1}$	11	111	-2	$\bar{2}$	110
0	$\bar{2}$	00	110	-2	$\bar{2}$	110
$\bar{1}$	$\bar{2}$	11	110	-3	-3	001

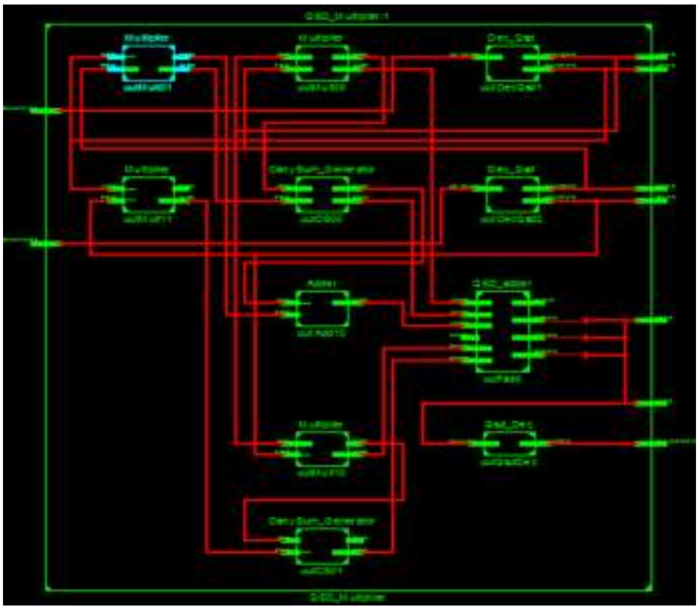


Figure 6: RTL Schematic of 2 Digit QSD Multiplier

VI. SIMULATION AND RESULTS

The QSD multiplier written in VHDL and synthesized. The results of the implemented QSDmultiplication operations were collected from the timing simulation of the Xilinx 13.1 software. The correctness of the results is confirmed.

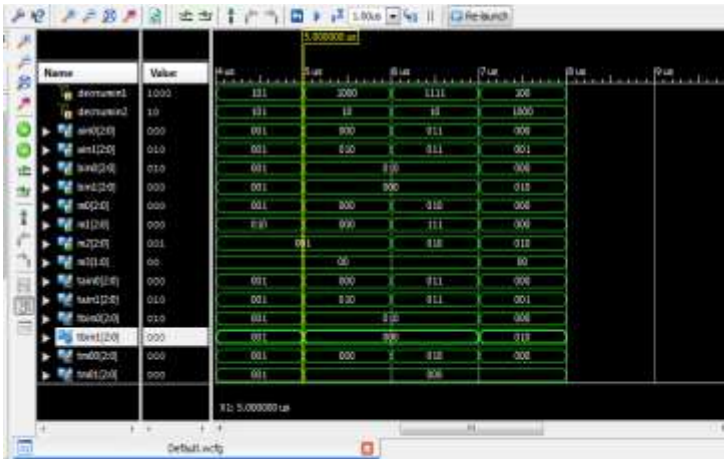


Figure7 : Simulated Result of QSD Multiplier

VII. CONCLUSIONS

The simulation of QSD multiplication are presented. The test confirms the superior performance of the QSD multiplier. With the QSD multiplication scheme, some well-known arithmetic algorithms can be directly implemented, In future still lower power dissipation can be achieved without modifying and degrading the circuit functionality. Consequently this QSD multiplier can be used as a building block for all arithmetic operations. It can be applied for construction of a high performance multiprocessor.These high performance multipliers are essential in digital processors.

Parameters	**Booth Multiplier**	**QSD Multiplier**
Time delay	**50.866 ns**	**7.243 ns**
Power Consumption	**13.25 W**	**0.060 W**

Above comparison shows that, these circuits consume less energy and power, and shows better performance also the delays of the proposed multiplier is less than the conventional binary multiplier.

REFERENCES

[1] **Kothuru.Ram Kumar, B.Praveen Kumar**, "*Fast Addition Using QSD VLSI Adder for Better Performance",* International Journal of Trend in Research and Development, Volume 2(6), ISSN: 2394-9333 www.ijtrd.com.

[2] **Bhavya Sree Kotte, S Saleem Malik,** , "*Design of Quaternary Signed Digit Adder",* IJSRD - International Journal for Scientific Research & Development| Vol. 3, Issue 06, 2015 | ISSN (online): 2321-0613.

[3] **S.Jakeer Hussain , K. Sreenivasa Rao** , *"Design and Implementation of Fast Addition Using QSD for Signed and Unsigned Numbers"* , International Journal of Engineering Research Volume No.3 Issue No: Special 2, pp: 52-54 ISSN:2319-6890)(online),2347-5013(print) 22 March 2014.

[4] **Jyoti R Hallikhed, Mahesh R.K**, "*VLSI Implementation of Fast Addition Using Quaternary Signed Digit Number System"*, International Journal of Ethics in Engineering & Management Education (ISSN: 2348-4748, Volume 2, Issue 5, May 2015).

[5] **Leela Bhardwaj Reddy, V. Narayana Reddy** *"VLSI Implementation of Fast Addition Subtraction and Multiplication (Unsigned) Using Quaternary Signed Digit Number System"* International journal & magazing of Engineering, Technology, Management and Research.

[6] **Tanay Chattopadhyay and Tamal 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, December 2012.