Quaternary Adder Design Using VHDL

Prashant Y. Shende[*], **Dr. R. V. Kshirsagar**[**]

[*] (M.Tech. Student, Deptt of Electronics (VLSI), PCE, Nagpur, India)
[**] (Professor & Head, Deptt. of Electronics (VLSI), PCE, Nagpur, India)

ABSTRACT

The high speed digital circuits became more prominent with incorporating information processing and computing. Arithmetic circuits play a very critical role in both general-purpose and application specific computational circuits. The modern computers lead to the deterioration in performance of arithmetic operations such as addition, subtraction, multiplication, division, on the aspects of carry propagation delay, large circuit complexity and high power consumption.

Designing this adder using QSD number representation allows fast addition/subtraction which is capable of carry free addition and borrows free subtraction because the carry propagation chain are eliminated, hence it reduce the propagation time in comparison with radix 2 system. QSD number system based on quaternary system, each digit can be represented by a number from -3 to -3. Operation on large number of digits such as 64, 128 or more, can be implemented with constant delay and complexity.

Keywords - QSD, carry/borrow free addition/ subtraction.

I. INTRODUCTION

Arithmetic operations are widely used and play important role in various digital systems such as computers and signal processors. Designing this Arithmetic unit using QSD number representation has attracted the interest of many researchers. Additionally, recent advances in technologies for integrated circuits make large scale arithmetic circuits suitable for VLSI implementation. In this paper, we propose a high speed QSD adder which is capable of carry free addition, borrow free subtraction. The QSD addition/subtraction operation employs a fixed number of minterms for any operand size. In QSD number system carry propagation chain are eliminated which reduce the computation time substantially, thus enhancing the speed of the machine. Signed digit number system offers the possibility of carry free addition. QSD Adder / QSD Multiplier circuits are logic circuits designed to perform high-speed arithmetic operations. In QSD number system carry propagation chain are eliminated which reduce the computation time substantially, thus enhancing the speed of the machine. The paper is structured as follows: Section II presents the Signed digit number. In Section III the QSD number system is presented. Section IV presents basic concept for performing any operation in QSD, first convert the binary or any other input into quaternary signed digit. Section V presents Adder/Substractor Design. Section VI presents simulation results. Then we provide our conclusions in Section VI.

II. Signed Digit Number

Signed digit representation of number indicates that digits can be prefixed with a – (minus) sign to indicate that they are negative. Signed digit representation can be used to accomplish fast addition of integers because it can eliminate carriers.

$$
\begin{aligned}
(11\bar{2}2)_2 &= 1\times 2^3 + 1\times 2^2 - 2\times 2^1 + 1\times 2^0 \\
&= 8+4-4+2 \\
&= 10
\end{aligned}
$$

II. QSD NUMBER SYSTEM

QSD numbers are represented using 3-bit 2's complement notation. Each number can be represented by

$$D = \Sigma xi\, 4^i$$

Where xi can be any value from the set {3, 2, 1, 0, 1, 2,3} for producing an appropriate decimal representation. A QSD negative number is the QSD complement of QSD positive number i.e.3 = -3, 2 = -2, 1 = -1. For digital implementation, large number of digits such as 64, 128, or more can be implemented with constant delay. A high speed and area effective adders and multipliers can be implemented using this technique. For digital implementation, large number of digits such as 64, 128, or more can be implemented with constant delay. A higher radix based signed digit number system, such as quaternary signed digit (QSD) number system, allows higher information storage density, less complexity, fewer system components and fewer cascaded gates and operations. A high speed and area effective adders and multipliers can be implemented using this technique [2, 1].

Also we can obtain redundant multiple representation of any integer Quantity using this QSD number system [3].

Examples of n digit QSD number are as follows: $2\bar{3}\bar{1}0, \bar{2}\bar{1}01, 3\bar{2}11, 310\bar{1}102\bar{3}$ etc.

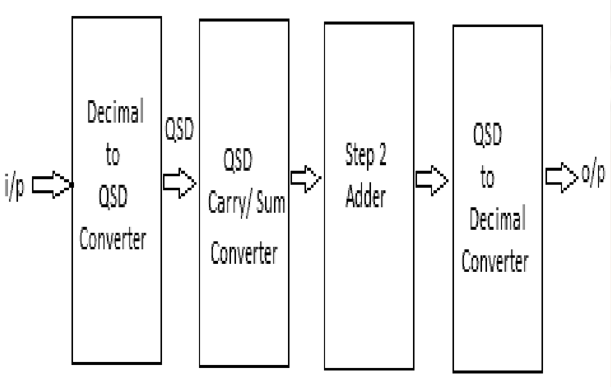
For example:

$$(13\bar{3}2)_{QSD} = 1\times 4^3 + 3\times 4^2 - 3\times 4^1 + 2\times 4^0$$
$$= 64+48-12+2$$
$$= (102)_{10}$$

The basic quaternary operators are very similar to binary operators and they are obtained from Boolean algebra.

IV. Basic Concept

For performing any operation in QSD, first convert the binary or any other input into quaternary signed digit



V. Adder/Substractor Design.

In arithmetic operation of digital computation addition is the most important operation. A carry-free addition is desirable as the number of digits is large. The carry-free addition can achieve by exploiting redundancy of QSD number and QSD addition. The redundancy allows multiple representations of any integer quantity i.e., 610 = 12QSD = 22QSD. There are two steps involved in the carry-free addition. The first step generates an intermediate carry and sum from the addend and augends. The second step combines the intermediate sum of the current digit with the carry of the lower significant digit. To prevent carry from further rippling, we define two rules. The first rule states that the magnitude of the intermediate sum must be less than or equal to 2. The second rule states that the magnitude of the carry must be less than or equal to 1.Consequently, the magnitude of the second step output cannot be greater than 3 which can be represented by a single-digit QSD number; hence no further carry is required. In step 1, all possible input pairs of the addend and augends are considered [4]. The output ranges from -6 to 6 as shown in Table 1.

Table 1.
The outputs of all possible combinations of a pair of addend (A) and augend (B).

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

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.Some numbers have multiple representations, but only those that meet the defined rules are chosen. The chosen intermediate carry and sum are listed in the last column of Table 2. 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. Since the intermediate carry is always between -1 and 1, it requires only a 2-bit binary representation. Finally, five 6-variable Boolean expressions can be extracted. The intermediate carry and sum circuit is shown in Figure 1.

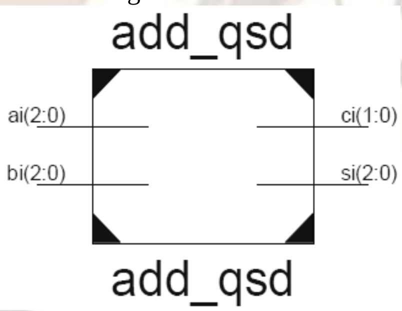


Figure 1. The intermediate carry and sum generator.

Table 2.
The intermediate carry and sum between -6 to 6.

SUM	QSD represented Number	QSD coded number
-6	$\bar{1}\,\bar{2},\bar{2}\,2$	$\bar{1}\,\bar{2}$
-5	$\bar{1}\,\bar{1},\bar{2}\,3$	$\bar{1}\,\bar{1}$
-4	$\bar{1}\,0$	$\bar{1}\,0$
-3	$0\,\bar{3},\bar{1}\,1$	$\bar{1}\,1$
-2	$0\,\bar{2},\bar{1}\,2$	$0\,\bar{2}$
-1	$0\,\bar{1},\bar{1}\,3$	$0\,\bar{1}$
0	00	00
1	$1\,\bar{3}$,01	01
2	$1\,\bar{2}$,02	02
3	$1\,\bar{1}$,03	$1\,\bar{1}$
4	10	10
5	$2\ \bar{3}$,11	11
6	$2\,\bar{2}$,12	12

In step 2, 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-3 shows all possible combinations of the summation between the intermediate carry and the sum.

Table 3.
The mapping between the inputs and outputs of the intermediate carry and sum.

INPUT				OUTPUT				
QSD		Binary		Decimal	QSD		Binary	
Ai	Bi	Ai	Bi	Sum	Ci	Si	Ci	Si
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	-1	01	111
2	1	010	001	3	1	-1	01	111
3	0	011	000	3	1	-1	01	111
0	3	000	011	3	1	-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	-1	011	111	2	0	2	00	010
-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	-1	010	111	1	0	1	00	001
-1	2	111	010	1	0	1	00	001
3	-2	011	110	1	0	1	00	001
-2	3	110	011	1	0	1	00	001
0	0	000	000	0	0	0	00	000
1	-1	001	111	0	0	0	00	000
-1	1	111	001	0	0	0	00	000
2	-2	010	110	0	0	0	00	000
-2	2	11	010	0	0	0	00	000
-3	3	101	011	0	0	0	00	000
3	-3	011	101	0	0	0	00	000
0	-1	000	111	-1	0	-1	00	111
-1	0	111	000	-1	0	-1	00	111
-2	1	110	001	-1	0	-1	00	111
1	-2	001	110	-1	0	-1	00	111
-3	2	101	010	-1	0	-1	00	111
2	-3	010	101	-1	0	-1	00	111
-1	-1	111	111	-2	0	-2	00	110
0	-2	000	110	-2	0	-2	00	110
-2	0	110	000	-2	0	-2	00	110
-3	1	101	001	-2	0	-2	00	110
1	-3	001	101	-2	0	-2	00	110
-1	-2	111	110	-3	-1	1	11	001
-2	-1	110	111	-3	-1	1	11	001
-3	0	101	000	-3	-1	1	11	001
0	-3	000	101	-3	-1	1	11	001
-3	-1	101	111	-4	-1	0	11	000
-1	-3	111	101	-4	-1	0	11	000
-2	-2	110	110	-4	-1	0	11	000
-3	-2	101	110	-5	-1	-1	11	111
-2	-3	110	101	-5	-1	-1	11	111
-3	-3	101	101	-6	-1	-2	11	110

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

The result of addition in this step ranges from -3 to 3. Since carry is not allowed in this step, the result becomes a single digit QSD output. The inputs, the intermediate carry and sum, are 2-bit and 3-bit binary respectively. The output is a 3-bit binary represented QSD number. The mapping between the 5- bit input and the 3-bit output is shown in Table 5.

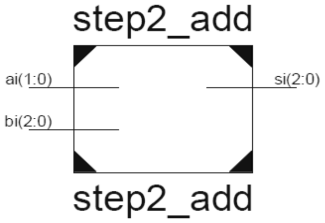


Figure 2. The second step QSD adder

Three 5- variable Boolean expressions can be extracted from Table 4. Figure 2 shows the diagram of the second step adder. The implementation of an *n*-digit QSD adder requires *n* QSD carry and sum generators and *n*-1 second step adder as shown in Figure 3. The result turns out to be an *n*+1-digit number [4].

Table-5:
The mapping between inputs and outputs of the second step QSD adder

INPUT				OUTPUT		
QSD		Binary		Decimal	QSD	Binary
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
-1	2	11	010	1	1	001
0	0	00	000	0	0	000
1	-1	01	111	0	0	000
-1	1	11	001	0	0	000
0	-1	00	111	-1	-1	111
-1	0	11	000	-1	-1	111
1	-2	01	110	-1	-1	111
-1	-1	11	111	-2	-2	110
0	-2	00	110	-2	-2	110
-1	-2	11	110	-3	-3	001

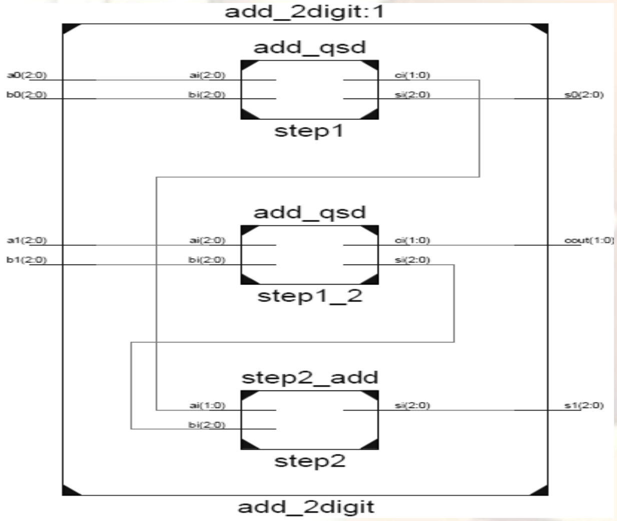


Figure 3. Four digit QSD adder.

VI. SIMULATION RESULTS

The QSD adder written in VHDL, compiled and simulation using modelsim. The QSD adder circuit simulated and synthesized. The simulated result for QSD adders as shown in figure-4.

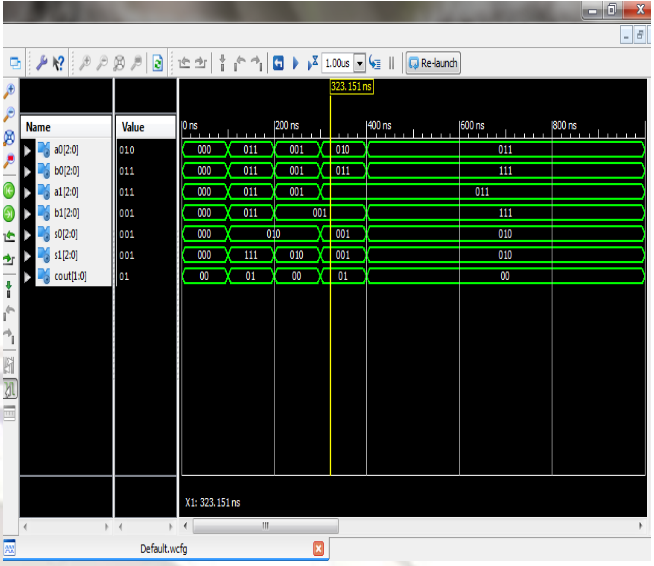

1.00us
Re-launch
323.151 ns
Name
Value
0 ns
200 ns
400 ns
600 ns
800 ns
a0[2:0] 010 000 011 001 010 011
b0[2:0] 011 000 011 001 011 111
a1[2:0] 011 000 011 001 011
b1[2:0] 001 000 011 001 111
s0[2:0] 001 000 010 001 010
s1[2:0] 001 000 111 010 001 010
cout[1:0] 01 00 01 00 01 00
X1: 323.151 ns
Default.wcfg

Figure 4: Simulated result QSD adder

VII. CONCLUSION

The proposed QSD adder is better than other binary adders in terms of number of gates and higher number of bits addition within constant time. Efficient design for adder block to perform addition or multiplication will increase operation speed. QSD number uses less space than BSD to store number; higher number of gates can be tolerated for further improvement of QSD adder.

REFERENCES

[1] Nagamani A. N, Nishchai S, “Quaternary High Performance Arithmetic Logic Unit Design”, *14th Euromicro Conference on Digital System Design 2011 IEEE.*

[2] Reena Rani, Laxmikant Singh, Neelam Sharma, “FPGA Implementation of fast Adder using Quaternary Sign Digit Number System”, *2009 International comferance on Trend in Eletronic and Photonic Deice & Systems (EECTRO-2009).*

[3] Pranali S.Kamble & S.M.Choudhary, “Review of VHDL Implementation of Quaternary Signed Adder System”, *International Journal on Advanced Electrical and Electronics Engineering, (IJAEEE), ISSN (Print): 2278-8948, Volume-1, Issue-1, 2012*

[4] Songpol Ongwattanakul Phaisit Chewputtanagul, David J. Jackson, Kenneth G. Ricks, “Quaternary Arithmetic Logic Unit on a Programmable Logic Device”, *proc. IEEE conference, 2001.*