Quaternary Arithmetic Logic Unit Design Using VHDL

Prashant Y. Shende[1], **R. V. Kshirsagar**[2]

Student, M. Tech. Department of Electronics (VLSI), PCE, Nagpur, India [1]

Professor and Head, Department of Electronics (VLSI), PCE, Nagpur, India[2]

Abstract: ***The arithmetic logic unit (ALU) is the core of a CPU in a computer; the adder cell is the elementary unit of an ALU. The constraints the adder has to satisfy are area, power and speed requirements. The delay in an adder is dominated by the carry chain. Arithmetic operations such as addition, subtraction and multiplication still suffer from known problems including limited number of bits, propagation time delay, and circuit complexity. Carry free arithmetic operations can be achieved using a higher radix number system such as quaternary Signed Digit (QSD). We proposed fast arithmetic logical unit based on quaternary signed digit number system where the carry propagation chain are eliminated, hence it reduce the propagation time in comparison with radix 2 system, each digit can be represented by a number from -3 to -3. In any n digit QSD number, each digit can be represented by a number from the digit set [-3,-2,-1, 0, 1, 2, 3]. Operations on a large number of digits can be implemented with constant delay and less complexity.***

Keywords: quaternary sign digit, fast computation, multiplier, quaternary logic, Carry/borrow free.

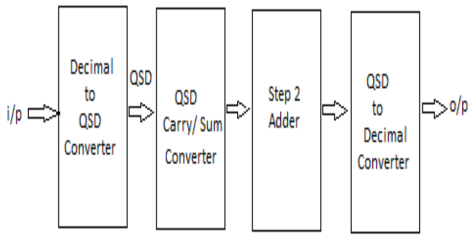
1. Introduction

The various digital systems such as computers and signal processors, arithmetic operation plays important role. The speed of system increases with increasing the speed of addition and multiplication. In conventional binary number system, carry may propagate all the way from the least significant digit to the most significant. Thus the addition time is dependent on the word length. In this paper, we propose a high speed QSD arithmetic logic unit which is capable of carry free addition, borrow free subtraction. The QSD addition/subtraction operation uses 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. 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, thus enhancing the speed of the machine. The paper is structured as follows: Section II presents the basic concept for performing any operation in QSD, first convert the binary or any other input into quaternary signed digit.

Section III presents Adder/Substractor Design. Section IV presents multiplier unit, logical unit is present in section V, in section VI present simulation results. Then we provide our conclusions in Section VII.

2. Basic Concept

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



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. 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: $13\bar{3}$.
For example:

$$(13\bar{3})_{QSD} = 1\times4^{2} + 3\times4^{1} - 3\times4^{0}$$

$$= 16 + 12 - 3 = (25)_{10}$$

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

3. Adder/ Subtracter Design

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, 6] the output ranges is from -6 to 6.

The range of the output is from -6 to 6 which represent the intermediate carry and sum in QSD format, some numbers have multiple representations, but only those that meet the defined rules are chosen. Both inputs and outputs can be encoded in 3-bit 2's complement binary number. Since the intermediate carry is always between -1 and 1, it requires only a 2-bit binary representation [1, 4]. Finally, five 6-variable Boolean expressions can be extracted. The intermediate carry and sum circuit is shown in Figure 1.

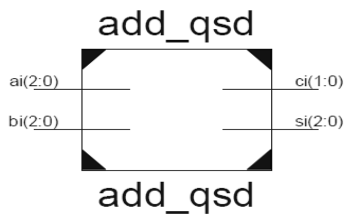


Figure 1: The intermediate carry and sum generator.

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. The Table 1 shows the outputs of all possible combinations of a pair of intermediate carry (A) and sum (B).

Table 1: The outputs of all possible combinations of a pair of intermediate carry (A) and sum (B)

B A	-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 2.

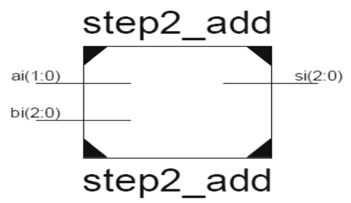


Figure 2: The second step QSD adder

Three 5-variable Boolean expressions can be extracted from Table 1 figure 2 shows the diagram of the second step added.

Table 2: The mapping between inputs and outputs of the second step QSD adder

INPUT				OUTPUT		
QSD		Binary		Decimal	QSD	Binary
Ai	Bi	Ai	Bi	Sum	Si	Si
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

The implementation of an n-digit QSD adder requires n QSD carry and sum generators and n-1 second step added as shown in Figure 3. The result turns out to be an n+1digit number [4].

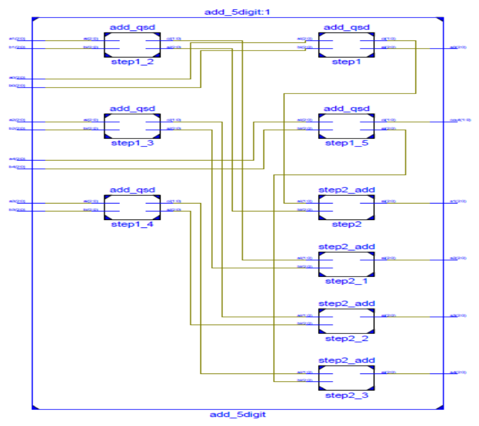


Figure 3: Five bit QSD adder

4. Multiplier Design

There are generally two methods for a multiplication operation: parallel and iterative. QSD multiplication can be implemented in both ways, requiring a QSD partial product generator and QSD adder as basic components. A partial product, Mi, is a result of multiplication between an n-digit input, An-1-A0, with a single digit input, Bi, where i = 0….n-1. The primitive component of the partial product generator is a single-digit multiplication unit whose functionality can be expressed as shown in Table 3.

Table 3: The outputs of all possible combinations of a pair of multiplicand (A) and multiplier (B)

BA	-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 single-digit multiplication produces 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, is between -2 and 2. According to Table 5, and using the same procedure as in creating Table 1 and 2, the mapping between the 6-bit input, A and B, to the 6-bit output, M and C, results in six 6-varible Boolean expressions which represent a single-digit multiplication operation. The diagram of a single-digit QSD multiplier is shown in Figure 4.

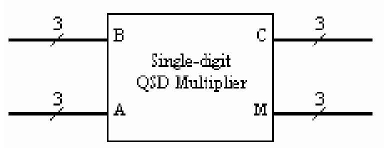


Figure 4: A Single- digit QSD Multiplier

The implementation of an n-digit partial product generator uses n units of the single-digit QSD multiplier. Gathering all the outputs to produce a partial product result presents a small challenge. The QSD representation of a single digit multiplication output, shown in Table 4, contains a carry-out of magnitude 2 when the output is either -9 or 9. This prohibits the use of the second step QSD adder alone as a gatherer [4, 5]. In fact, we can use the complete QSD adder from the previous section as the gatherer. Furthermore, the intermediate carry and sum circuit can be optimized by not considering the input of magnitude 3. The QSD partial product generator implementation is shown in Figure 5.

Table 4: The QSD representation of a single-digit multiplication output

SUM	QSD represented Number	QSD coded number
-9	$\bar{2}\bar{1}\bar{3}3$	$\bar{2}\bar{1}$
-6	$\bar{1}\bar{2},\bar{2}2$	$\bar{1}\bar{2}$
-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
6	$2\bar{2}$,12	12
9	$21,3\bar{3}$	21

An nxn-digit QSD multiplication requires n partial product terms. In an iterative implementation, a 2ndigit QSD adder is used to perform add-shift operations between the partial product generator and the accumulator. After n iterations, the multiplication is complete. In contrast, a parallel implementation requires n partial product circuits and n-1 QSD adder units.

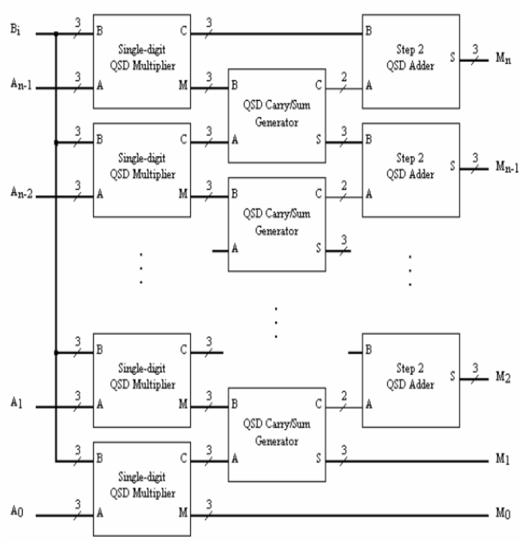


Figure 5: The n-digit QSD partial product generator

A binary reduction sum is applied to reduce the propagation delay to 0 (log n). The schematic of a 4x4 parallel QSD multiplication is shown in Figure 6.

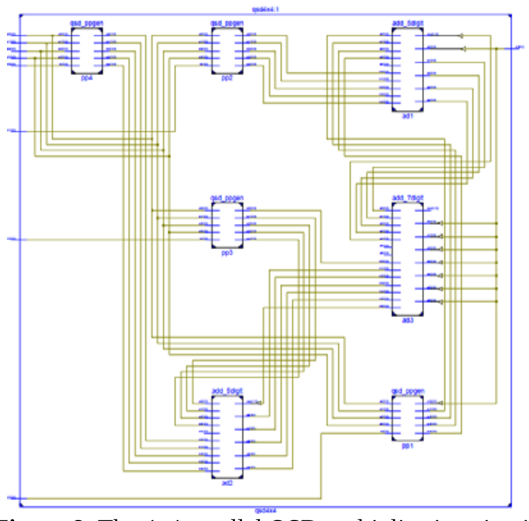


Figure 6: The 4x4 parallel QSD multiplication circuit

5. Logic Unit Design

The quaternary logic unit includes the inverter, MAX function implementation and MIN function implementation which are similar to the OR function and AND function respectively. The quaternary logic levels are expressed in terms 2-bit binary and same binary AND and OR functions can be used to realiase MIN and MAX function respectively. The truth table for inverter, MAX, MIN function is shown in table-5.

Table 5: Truth Table of Inverter, Max function, Min function

Inverter	
In	Out
0	3
1	2
2	1
3	0

MAX					
		A			
		0	1	2	3
	0	0	1	2	3
	1	1	1	2	3
	2	2	2	2	3
B	3	3	3	3	3

MIN					
		A			
		0	1	2	3
	0	0	0	0	0
	1	0	1	1	1
	2	0	1	2	2
B	3	0	1	2	3

6. Results

The quaternary based arithmetic unit is written in VHDL and synthesized on SPARTAN 3(3 xc3s50-5pq208) FPGA device using Xilinx project navigator tool. Table 6 shows the LUT utilization and delay performance of the QSD ALU. The delay comparison graph is shown in figure 7 and figure 8 shows the number of LUT used by the arithmetic units. The QSD number is capable of representing twice as much magnitude in each digit compared to the binary representation. The overall delay of the QSD ALU is total of conversion delay and computation delay.

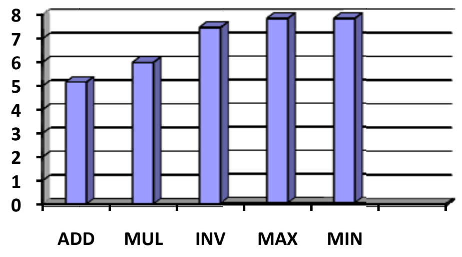


Figure 7: Graph showing timing delay of arithmetic units

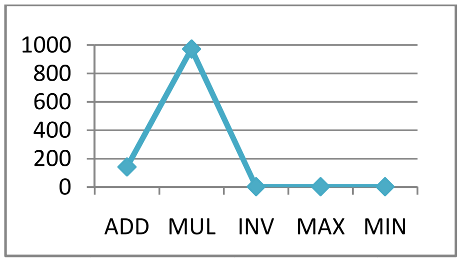


Figure 8: The number of LUT used by the arithmetic units.

Table 6: Timing delay and number of LTU used in SPARTAN 3

Opcode	*Alu unit*	*Delay*	*LTU utilization*
0	ADD	5.146	142
1	MULTIPLY	5.98 ns	972
10	INVERTER	7.665 ns	2
11	MAX/OR	7.842 ns	2
100	MIN/AND	7.842 ns	2

7. Conclusion

The proposed QSD arithmetic logic unit is better than other binary arithmetic logic unit in terms of number of gates and higher number of bits operation within constant time. Efficient design for adder block to perform addition or multiplication will increase operation speed. QSD number uses 25% 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.

[2] Reena Rani, Laxmikant Singh, Neelam Sharma, "FPGA Implementation of fast Adder using Quaternary Sign Digit Number System", 2009 International conference on Trend in Electronic 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", Electrical and Computer Engineering The University of Alabama, Tuscaloosa, AL 35487-0286, USA.

[5] Ifat Jahangir, Dihan Md. Nuruddin Hasan, Nahian Alam Siddiquet , Shajid lslarn, Md. Mehedi Hasan, "Design of Quaternary Sequential Circuits Using a Newly Proposed Quaternary Algebra", Proceedings of 2009 12th International Conference on Computer and Information Technology, (ICCIT 2009) 21-23 December, 2009, Dhaka, Bangladesh

[6] Vasundara Patel K S, K S Gurumurthy, "Design of High Performance Quaternary Adders", International Journal of Computer Theory and Engineering, Vol.2, No.6, December, 2010, 1793-8201