International Journal of Research Studies in Science, Engineering and Technology
Volume 1, Issue 9, December 2014, PP 144-148
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

Implementation of Quaternary Signed Adder System

Vadathya Mohan[*], K. Madan Mohan[**]

[*]PG Scholar, Electronics and Communication Engineering, Intel Engineering College, AP, India
[**]Assistant Professor, Electronics and Communication Engineering, Intel Engineering College, AP, India

Abstract: *With the binary number system, the computation speed is limited by formation and propagation of carry especially as the number of bits increases. Using a quaternary Signed Digit number system one may perform carry free addition, borrow free subtraction and multiplication. However the QSD number system requires a different set of prime modulo based logic elements for each arithmetic operation. A carry free arithmetic operation can be achieved using a higher radix number system such as Quaternary Signed Digit (QSD). In QSD, each digit can be represented by a number from -3 to 3. 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.*

1. INTRODUCTION

These high performance adders are essential since the speed of the digital processor depends heavily on the speed of the adders used is the system. Also, it serves as a building block for synthesis of all other arithmetic operations. Adders are most commonly used in various electronic applications e.g. Digital signal processing in which adders are used to perform various algorithms like FIR, IIR etc. In past, the major challenge for VLSI designer is to reduce area of chip by using efficient optimization techniques. Then the next phase is to increase the speed of operation to achieve fast calculations like, in today's microprocessors millions of instructions are performed per second. Speed of operation is one of the major constraints in designing DSP processors [11]. The redundancy associated with signed-digit numbers offers the possibility of carry free addition. The redundancy provided in signed-digit representation allows for fast addition and subtraction because the sum or difference digit is a function of only the digits in two adjacent digit positions of the operands for a radix greater than 2, and 3 adjacent digit positions for a radix of 2. Thus, the add time for two redundant signed-digit numbers is a constant independent of the word length of the operands, which is the key to high speed computation. The advantage of carry free addition offered by QSD numbers is exploited in designing a fast adder circuit. Additionally adder designed with QSD number system has a regular layout which is suitable for VLSI implementation which is the great advantage over the RBSD adder. An Algorithm for design of QSD adder is proposed. This algorithm is used to write the verilog code for QSD adders. verilog codes for QSD adder is simulated and synthesized and the timing report is generated. The timing report gives the delay time produced by the adder structure.

Binary signed-digit numbers are known to allow limited carry propagation with a somewhat more complex addition process requiring very large circuit for implementation [4] [10]. A special higher radix-based (quaternary) representation of binary signed-digit numbers not only allows carry-free addition and borrow-free subtraction but also offers other important advantages such as simplicity in logic and higher storage density [15].

2. QUATERNARY SIGNED DIGIT NUMBERS

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

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

where Xi can be any value from the set {-3,-2,-1,,0,1,2,3}for producing an appropriate decimal representation. 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].

3. DECIMAL NUMBER TO QSD NUMBER CONVERTER

We review some ideas related to this decimal number to QSD number converter based on the text book. We can achieve making of such kind of algorithm which aids to convert any n digit decimal number into higher radix QSD number system. For the same purpose we exploit some fundamentals as we are most familiar with those are as given below, and these fundamental plays an important role in corresponding algorithm. The numbers that are positive are called unsigned, and numbers that can also be negative are called signed. Here, two types of binary numbers are defined.1.Unsigned binary number consist of only magnitude and this number is always positive.2.Signed binary number consists of magnitude as well as sign. We can use both unsigned binary integer and signed binary integer depending on the requirement of inputs given to algorithm. In the decimal number system the sign of a number is indicated by a + or _ symbol to the left of the most- significant digit. In binary number system the sign of a number is denoted by the left most bit. The left-most bit is equal to 0 for a positive number and for a negative number it is equal to 1.Therefore,in signed numbers the left most bit represents the sign, and the remaining n_1 bits represent the magnitude, as illustrated in fig 1.It is important to note the difference in the location of most-significant bit(MSB). In unsigned numbers the magnitude of a number is represented by all bits; hence all n bits are significant in defining the magnitude. Therefore, the MSB is the leftmost bit,b_{n-1} Since the left most bit of unsigned integer is called the MSB but left most bit of signed integer is not MSB and in signed numbers there are n_1 significants bits,and the MSB is in bit position b_{n-1} as illustrated in fig a and fig b respectively. This algorithm can be directly applied to both unsigned and 2's complement binary integer addition/subtraction. The algorithm depicts that decimal number is given as input in the form of n bit binary number as modelsim and Xilinx softwares takes the inputs in binary form. Algorithm states that it takes any n digit decimal number and converts into its equivalent Quaternary signed digit number wether the given input is positive decimal number or negative number. If decimal number is positive, then after conversion we get QSD number in which each digit is positive is found, and also each digit is represented with 3 bit 2's complement notation is displayed in simulation result. If decimal number is negative, then after conversion we get QSD number in which each digit is negative is found and also each digit is represented with 3 bit 2's complement notation is displayed in simulation result.

4. DESIGN ALGORITHM OF QSD ADDER

In QSD number system carry propagation chain are eliminated which reduce the computation time substantially, thus enhancing the speed of the machine [31]. As range of QSD number is from -3 to 3, the addition result of two QSD numbers varies from -6 to +6 [30]. Table I depicts the output for all possible combinations of two numbers. The decimal numbers in the range of -3 to +3 are represented by one digit QSD number. As the decimal number exceeds from this range, more than one digit of QSD number is required. For the addition result, which is in the range of -6 to +6, two QSD digits are needed. In the two digits QSD result the LSB digit represents the sum bit and the MSB digit represents the carry bit. To prevent this carry bit to propagate from lower digit position to higher digit position QSD number representation is used [37]. QSD numbers allow redundancy in the number representations. The same decimal number can be represented in more than one QSD representations. So we choose such QSD represented number which prevents further rippling of carry. To perform carry free addition, the addition of two QSD numbers can be done in two steps [4]:

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

Step 2: Second step combines intermediate sum of current digit with the intermediate carry of the lower significant digit.

So the addition of two QSD numbers is done in two stages. First stage of adder generates intermediate carry and intermediate sum from the input digits. Second stage of adder adds the intermediate sum of current digit with the intermediate carry of lower significant digit. To remove the further rippling of carry there are two rules to perform QSD addition 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.

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. The output ranges from –6 to 6 as shown in Table 1.

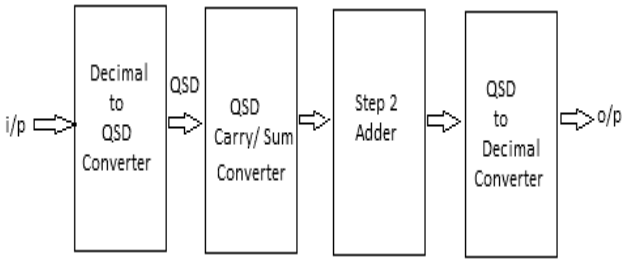
Table I. *The Intermediate Carry and Sum between -6 To +6*

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

According to these two rules the intermediate sum and intermediate carry from the first step QSD adder can have the range of -6 to +6. But by exploiting the redundancy feature of QSD numbers we choose such QSD represented number 655 which satisfies the above mentioned two rules. When the second step QSD adder adds the intermediate sum of current digit, which is in the range of -2 to +2, with the intermediate carry of lower significant digit, which is in the range of -1 to +1, the addition result cannot be greater than 3 i.e., it will be in the range of -3 to +3. The addition result in this range can be represented by a single digit QSD number; hence no further carry is required. In the step 1 QSD adder, the range of output is from -6 to +6 which can be represented in the intermediate carry and sum in QSD format as shown in table I. We can see in the first column of TableI that some numbers have multiple representations, but only those that meet the above defined two rules are chosen. The chosen intermediate carry and intermediate sum are listed in the last column of Table I as the QSD coded number.

5. BASIC CONCEPT

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

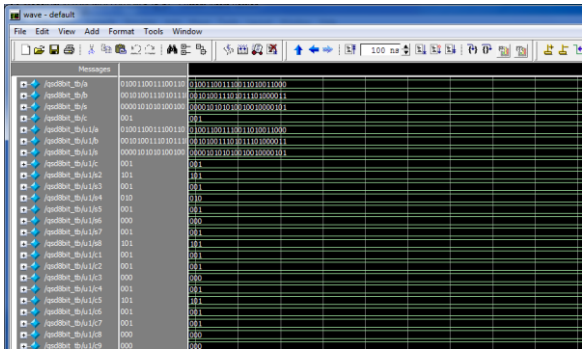


6. ADDER/SUBSTRACTOR DESIGN

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.

7. SIMULATION RESULTS

The QSD adder written in verilog, compiled and simulation using modelsim. The QSD adder circuit simulated and synthesized. The simulated result for QSD adders



8. 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.

[5] 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.

[6] F. Kharbash and G. M. Chaudhry, "Reliable Binary Signed Digit Number Adder Design", IEEE Computer Society Annual Symposium on VLSI, pp 479-484, 2007.

[7] John Moskal, Erdal Oruklu and Jafar Saniie, "Design and Synthesis of a Carry-Free Signed-Digit Decimal Adder", IEEE International symposium on Circuits and Systems, pp 1089-1092, 2007.

[8] Kai Hwang, "Computer Arithmetic Principles, Architecture and Design", ISBN 0-471-03496-7, John Wiley & Sons, 1979.

[9] P. K. Dakhole, D.G. Wakde, "Multi Digit Quaternary adder on Programmable Device : Design and verification", International Conference on Electronic Design, pp. 1-4, Dec 2008.

[10] Behrooz Parhami, "Carry-Free Addition of Recoded Binary Signed- Digit Numbers", IEEE Transactions on Computers, Vol. 37, No. 11, pp. 1470-1476, November 1988.