IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

DESIGN OF HYBRID QUATERNARY SIGNED DIGIT (QSD) BASED DIVIDER USING VHDL

Sandeep Kaur*, Sarabdeep Singh
* M.Tech. Student, Deptt of Electronics (VLSI), CEC, Landran, India).
(Assistant Professor, Deptt. of Electronics (VLSI), CEC, Landran, India).

ABSTRACT

Arithmetic operations in digital signal processing plays an important. While these operations suffer from many problems like propagation time delay, high power consumption and circuit complexity. QSD number system allows a fast addition, subtraction and multiplication. In this paper, we proposed a divider which is made by using QSD technique for fast division without carry propagating delay. In this division method the carry propagation chain are eliminated and reduces the propagation time delay. QSD number system based on quaternary system can be represented by a number from -3 to 3. The proposed design is developed using VHDL and implemented on Xilinx and results are showed on the ModelSim. The results are compared with the other dividers.

KEYWORDS: QSD- Quaternary signed digit, VHDL- VHSIC hardware description language.

INTRODUCTION

Arithmetic operation plays an important role in various digital systems such as computers and signal processors. Arithmetic operations are widely used. Recent advances in technologies for integrated circuits make large scale arithmetic circuits suitable for VLSI implementation, arithmetic operations using QSD number system has attracted the interest of many researchers. In this paper, we propose a QSD divider using QSD adder as a basic block. This QSD number system based divider is capable for carry free division and eliminates the carry propagation chains which reduce the computation time and enhances the speed of the circuit. Signed digit number system offers the possibility of carry free addition, borrow free subtraction and carry free divisions.

SIGNED DIGIT NUMBER

Signed digit number representations are prefixed with a – (minus) sign to indicate that they are negative numbers. Signed digit numbers used to accomplish fast addition, substraction, multiplication and division of integers because it can eliminate carry.

$(1122)_2 = 1 * 2^3 + 1 * 2^2 - 2 * 2^1 + 1 * 2^0$
$= 8 + 4 - 4 + 2$
$= 10$

QSD NUMBER SYSTEM

QSD numbers are represented by using 3-bit 2's compliment notation. Each number can be represented by
$D = \sum = x_i 4^i$

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

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

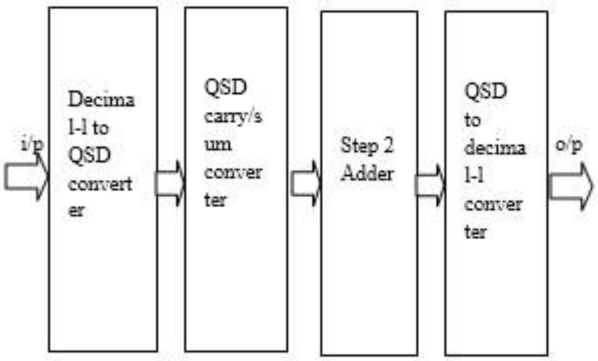
Example of an n digit QSD numbers are as follows:
$2\bar{3}\bar{1}0,\bar{2}\bar{1}01,3\bar{2}11,310\bar{1}102\bar{3}$ etc.
For example:

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

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

BASIC CONCEPT

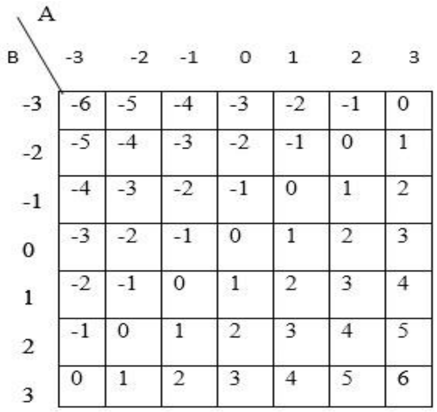
To perform any operation in QSD, first convert the binary or any other input into quaternary signed digit.



ADDER/SUBSTRACTOR DESIGN

In arithmetic operation of digital 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. Redundancy allows multiple representations of any integer i.e., 610 = 20QSD = 22QSD. Two steps are involved in carry free addition. First step generates an intermediate carry and sum from the addends and augends. Second step combines intermediate sum of the current digit with the carry of lower significant digit. To prevent carry from rippling, two rules are defined. First rule states that the magnitude of intermediate sum must be less than or equal to 2. In second rule, it states that the magnitude of 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 single digit QSD numbers; hence no further carry is required. In step 1, all the possible values of addends and augends are considered. The output ranges from -6 to 6 as shown in table1.Outputs of all possible combinations of addend (A) and (B).

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 Table1 is shown below with the addends and augends:



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

The range of outputs is from -6 to 6 which are shown in the form of intermediate carry and sum QSD format as shown in table2. Some numbers have multiple representations, but only those that meet the defined rules are chosen. The chosen intermediate carry and sum are listed in last column of table2. Both the inputs and outputs can be represented in 3bit 2's compliment binary number.

The mapping between inputs, addends and augends, and outputs, the intermediate carry and sum are shown in binary format of table3. The intermediate carry is always between -1 and 1. It requires only 2bit binary representation. Finally, five 6-variables Boolean expressions can be extracted. The intermediate carry and sum is shown in figure1.

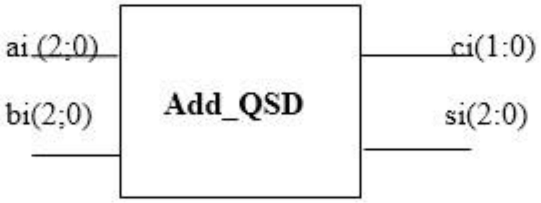


Table2: QSD numbers and coded numbers are shown below.

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	$\bar{1}0$	$0\bar{1}$
-3	$0\bar{3},\bar{1}1$	$1\bar{1}$
-2	$0\bar{2},\bar{1}2$	$2\bar{2}$
-1	$0\bar{1},\bar{1}3$	$1\bar{1}$
0	00	00
1	$1\bar{3}$,01	01
2	$1\bar{3}$,01	02
3	$1\bar{1}$,03	11
4	10	10
5	$2\bar{3}$,11	11
6	$2\bar{2}$,12	12

QSD DIVIDER

QSD divider is made by using QSD algorithm. In this divider the carry propagating chain are eliminated and the speed of the operation is increased. By QSD divider the fast division process is done. Steps of QSD division are first take two binary numbers for the division. One binary number is dividend and other is divisor. Then convert the dividend and divisor into QSD number system. After conversion, shift divisor right and compare it with current dividend. If divisor is larger, shift 0 as the next bit of the quotient. If divisor is smaller, subtract to get new dividend and shift 1 as the next bit of the quotient.

CLOCK ENABLE FOR ADDITION OF TWO COMPONENTS

In this section, clock is enabled for the addition of two components of two binary numbers.

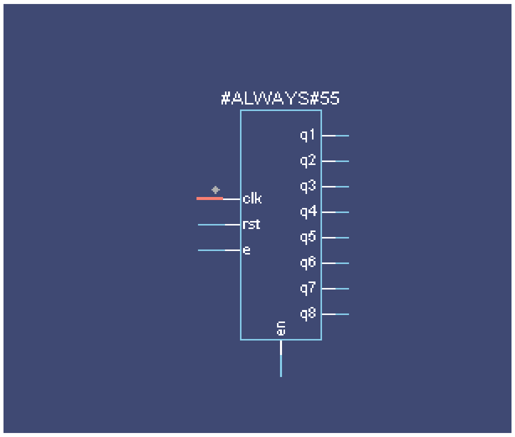


Figure2: clock enabling.

RESET BLOCK

In reset block the adders are reset and are able to do the further operations again.

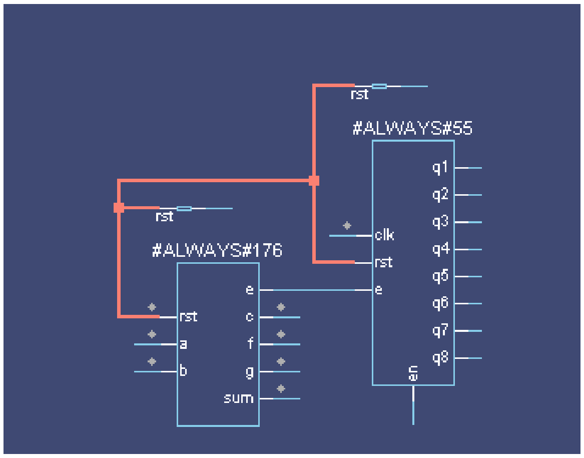


Figure3: Reset circuit.

FLIP FLOP STORAGE UNITS FOR INTERMEDDIATE DIVISION OPERATION

This is the storage unit in which the values of division are stored and this is storage unit to store the data of division.

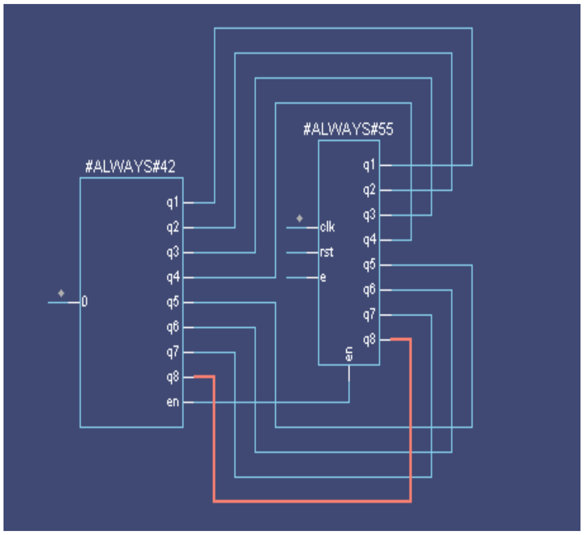


Figure4: Storage unit

ADDER PROCESS

Here the adder is shown in the figure 5. In which the addition process is done. The addition of two components or two QSD numbers can be done in it.

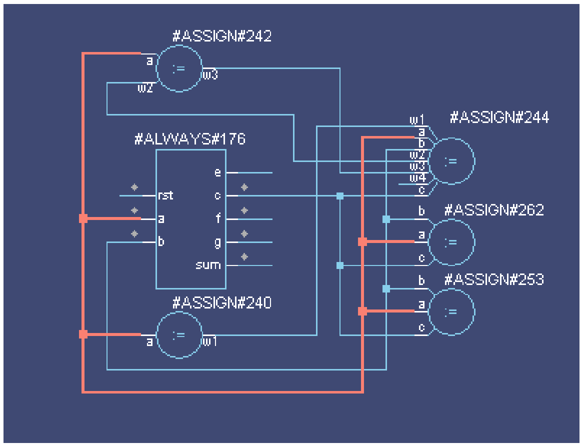


Figure5: Adder circuit.

DIVISION PROCESS

It is shown in the figure6 that the division of two components is done here. This is the division process.

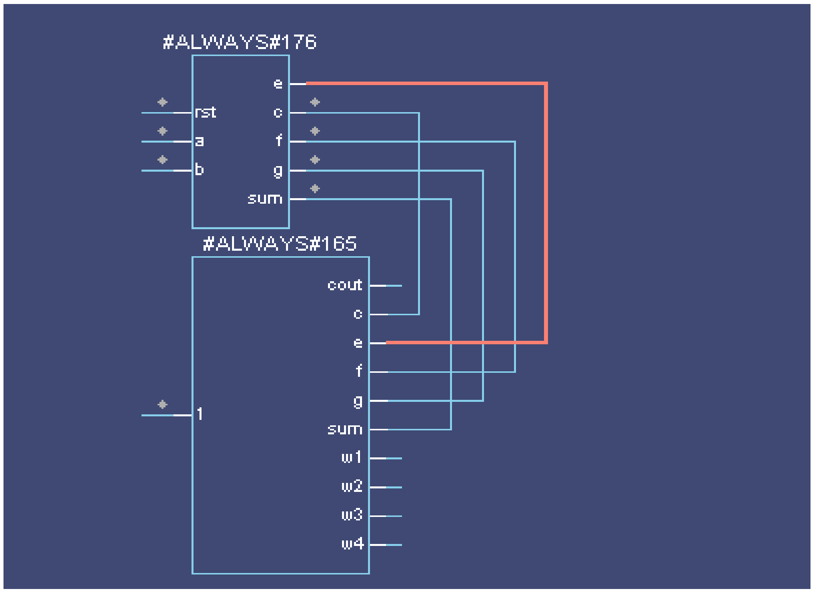


Figure6: Division circuit.

STORAGE VALUE FOR ADDED COMPONENETS REGISTERS

After division the data has to be stored, so that is stored in the register components.

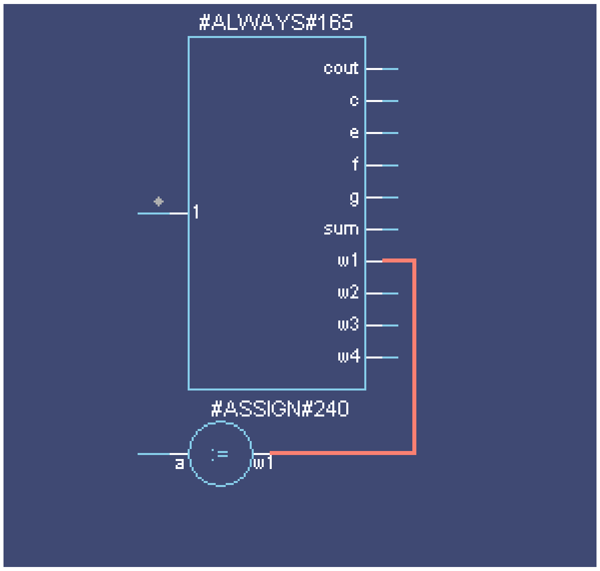
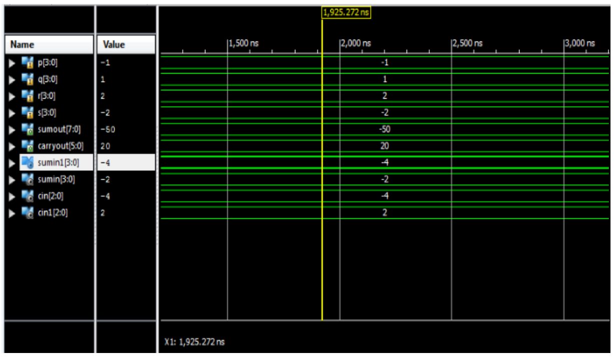


Figure7: Storage unit.

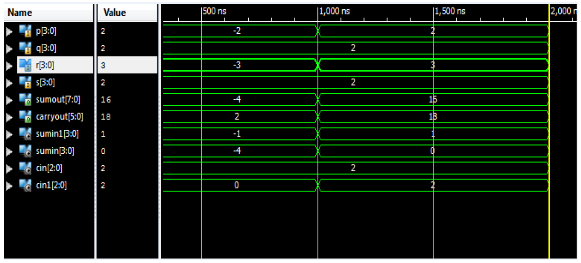
SIMULATION RESULTS

The QSD divider is written in VHDL, compiled and simulation using modelsim. The QSD divider circuit simulated and synthesized. The simulated result for QSD divider is shown in figure.



Name	Value
p[3:0]	-1
q[3:0]	1
r[3:0]	2
s[3:0]	-2
sumout[7:0]	-50
carryout[5:0]	20
sumin1[3:0]	-4
sumin[3:0]	-2
cin[2:0]	-4
cin1[2:0]	2

FIGURE8: Simulation of QSD adder.



Name	Value
p[3:0]	2
q[3:0]	2
r[3:0]	3
s[3:0]	2
sumout[7:0]	16
carryout[5:0]	18
sumin1[3:0]	1
sumin[3:0]	0
cin[2:0]	2
cin1[2:0]	2

Figure9: simulation of QSD divider.

CONCLUSION

The proposed QSD divider is better than other binary dividers in terms of number of gates and power. The power used by QSD divider is 23.50Mw and total delay is 18.06ns. Efficient design of divider block to perform division process will increase the operation speed. QSD number uses less space than BSD to store number; higher number of gates can be tolerated for further improvement of QSD divider.

REFRENCES

[1] Prashant y. shende , Dr. R. V. Kshirsagar, "Quaternary Adder Design Using VHDL" International journal of Engineering research and Appilications(IJERA),ISSN:2248-9622, Vol.- 3, Issue 3, May –jun 2013.

[2] Nagamani .A.N,"Quaternary High Performance Arithmetic Logic Unit Design", IEEE Conference on Digital System Design., pp.698–783, August 2011.

[3] "Architecture for Software Defined Cognitive Radio" A [6] Vasundara Patel, "Arithmetic Operations in Multi-valued Logic ," International Journal of VLSI Design and Communication Systems, Vol.1, Issue.1, pp.12-13, March 2010.

[4] 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.

[5] T. Chattopadhyay and J.N. Roy, "All-optical multi-valued computing: the future challenges and opportunities", International conference on convergence of Optics and Electronics, (COE 11), March 26-27, Kolkata, Pp. 94-101, 2011, ISBN 978-81-906401-1-4.

[6] T. Chattopadhyay, "All-optical symmetric ternary logic gate", Optics and Laser Technology, Vol. 42, Pp.1014-1021, 2010.

[7] 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.

[8] Etiemble, M. Israel, "Comparison of binary and Multivalued ICs according to VLSI criteria", IEEE Computer, Pp. 28-42, April 1988.

[9] S. L. Hurst, "Multiple-Valued Logic—Its Status and its Future". IEEE Transactions computers, Vol. C-33, No. 12, Pp. 1160-1179, 1984.

[10] H. Kerkhoff, M. Tervoert, "Multiple-Valued logic Charge-Coupled Devices". IEEE Transactions computers, Vol. C-30, No. 9, Pp. 644-652, 1981.

[11] D.D. Givone, M.E. Liebler and R.P. Roesser, “A method of solution for multiple valued logic expression”, IEEE Trans. on computers, short notes, Pp. 464-467, April 1971. D