International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 10, Number 1 (2017), pp. 53-61
© International Research Publication House
http://www.irphouse.com

Implementation of High Speed and Low Area Digital Radix-2 CSD Multipliers using Pipeline Concept

[1]M.Lakshmi Kiran and [2]Dr. K.Venkata Ramanaiah

[1]Ph.D. Scholar, and [2]Jr., *Member, IEEE,*

[1,2]Department of Electronics and Communication Engineering,
Y.S.R.E.C. of Yogi Vemana University, Proddatur-516360, India.

Abstract

Digital multipliers play a vital role in DSP based systems, DIP based systems, NN based systems and etc., Since it is required by most & many in almost all algorithms in the fields specified above and hence designing such digital multipliers with low power, high speed and low area is mandatory, so that the system is efficient in size, power, speed and cost. There are many types of algorithms or multipliers available in literature, Such as Modified Booth multiplier, Wallace tree multiplier, Radix-2 CSD multiplier and so on. Especially, there have been extensive researches about Radix-2 CSD multipliers as CSD numbers represented with lesser number of non-zero digits. In this paper, a study on various techniques of CSD conversions and best of these conversions is applied to different CSD multipliers. Horner method based multiplication is taken as reference in this paper. It reduces complexity. But, it faces the problem of high power consumption along with low speed. In this paper, Pipeline based multiplication is proposed which eliminates above said problems. This method saves execution time by 87.96% time and reduces number of slice LUTs by 92.86% in comparison with the earlier method (Horner based) while designing them in Verilog HDL and targeted on Xilinx Vertex-7 device.

Index Terms: CSD (Canonic Signed Digit), DSP (Digital Signal Processing), DIP (Digital Image processing), NN (Neural Network), HDL (Hardware Description Language).

I. INTRODUCTION

Digital Image Processing systems, Digital Signal Processing systems, Neural Network systems and related fields require digital multiplier [1]as a most basic building block, without which such design won't exist. Therefore efficient design of such Systems is majorly depends upon design of digital multipliers. There have been great researches on digital multipliers and related techniques are grouped mainly into 2 groups [1]. In one kind of digital multipliers, focus of research is on reducing number of processing steps (i.e number of adders) such as Wallace tree multiplier[8],[9], Column bypass multiplier, and Row bypass multiplier [11]. In the second kind of digital multipliers, focus of research is on reducing number of partial products (i.e number of non-zero digits in the number) such as Booth multiplier [1], Radix-2 CSD multiplier[1],[6],[7]. Among these, design of radix-2 CSD multiplier is considered in this paper as the central theme, as it is popularized in these days. Also, it would have minimum partial products, so that some major design resources are said to be saved. This leads to high speed, low area. And, hence researchers are focusing research on Radix-2 CSD multiplier [1],[6],[7]. In this paper, different techniques of CSD conversions and Horner CSD multiplication algorithm are studied [1],[6]. Also, Pipeline CSD multiplication is proposed. In section II different CSD conversion techniques and existed Horner CSD multiplication technique [1] are presented, in section III proposed pipeline CSD multiplication [3],[4] is discussed, in section IV comparisons of two conversion techniques and two CSD multiplication algorithms are presented.

II. CSD CONVERSION TECHNIQUES

CSD multiplication can be done in 2 steps. In the first step multiplier is converted into Canonic Signed Digit number from the binary number. In the second step actual multiplication will be done. Here 2 types of CSD conversion techniques are presented. They are Direct CSD conversion [1],[4],[5] and Bit by bit CSD conversion.

A. Direct CSD Conversion: - Unlike binary number system contains 1 and 0, Radix-2 CSD number contains 1, 0 and -1. In general radix-2N number system contains integers from –N to N. This is straight forward conversion from binary numbers to CSD number. For this conversion binary number is inputted to the following equations.[1],[4] and [5]

```
$a_{-1} = 0$, $y_{-1}= 0$,$a_W = a_{W-1}$
for (i= 0 to W-1)
{
$Q_i = a_i$ xor $a_{i-1}$
yi = $y_{i-1}$ and $Q_i$
```

```
    ti = $a_{i+1}$ and $y_i$
    $c_i$ = 1- $t_i$ - $t_i$
}
```

In the above algorithm input binary number is represented by ‘a’ and output CSD number is denoted by ‘c’ and ‘W’ represents length of the number.

B. Bit by Bit CSD Conversion:- Unlike in earlier technique, this technique is systematically modeled, and can be easily designed systematically in Verilog HDL. This conversion is done bit by bit. First of all, OBCSD (One Bit CSD) module is constructed as shown in Fig.1 which converts a binary bit to a CSD bit. Fig.2 represents the block diagram for 4 bit binary number to CSD number conversion. a[-1] is assumed to 0, a[4] is 0 for unsigned numbers and 1 for signed numbers. For an example if input a=0111, then output c=1001’.

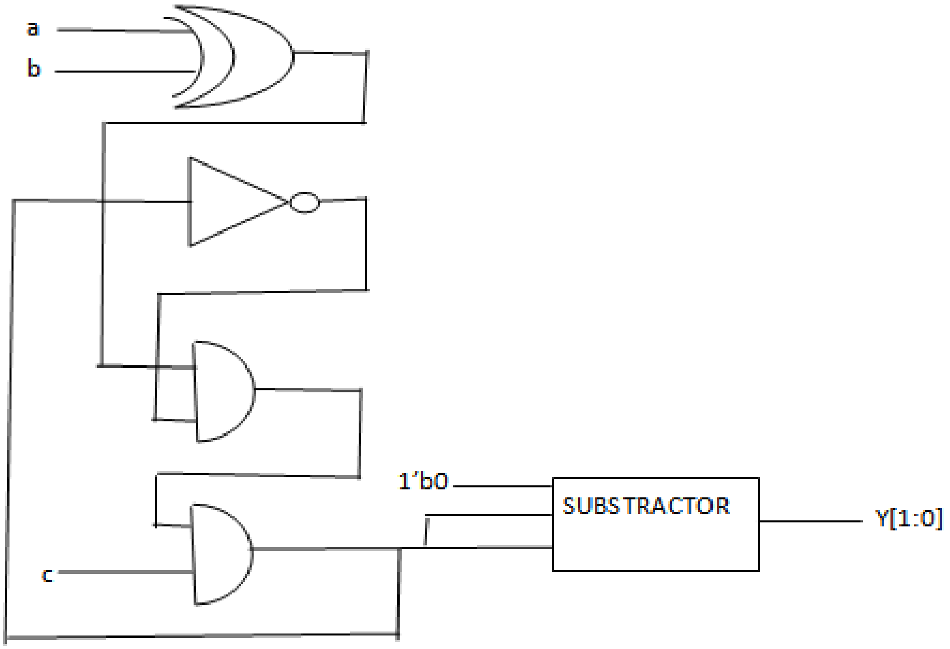

Fig.1:- A logical diagram for OBCSD(One Bit CSD module)

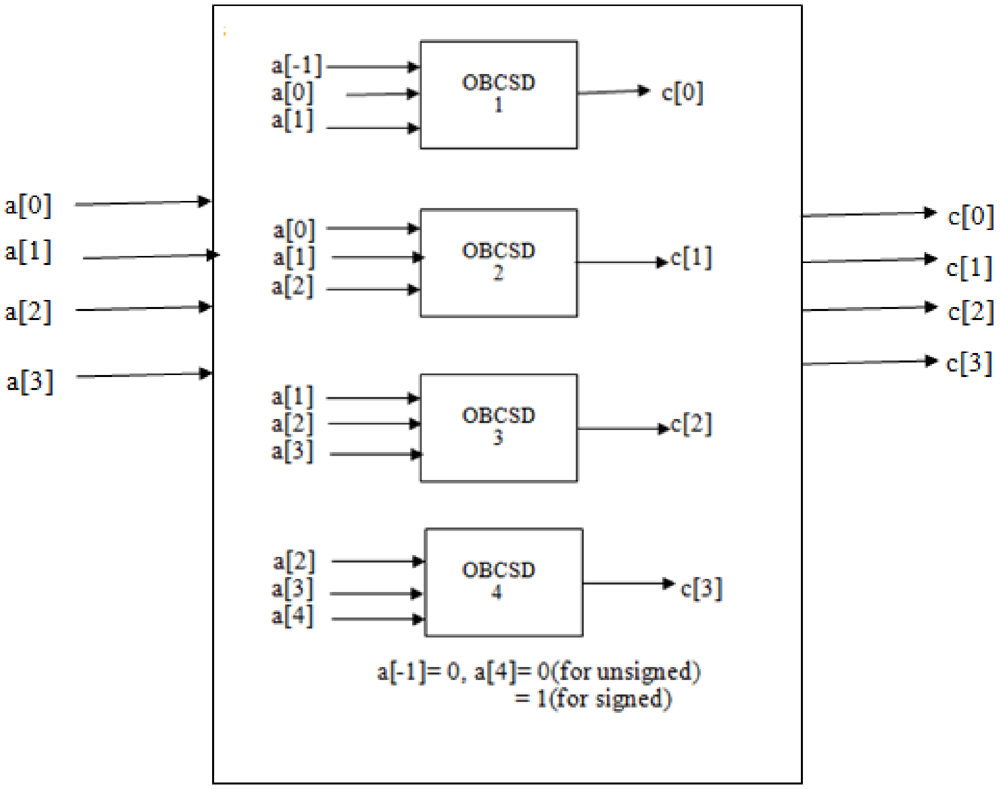

Fig.2:- a block diagram for converting 4 bit binary number to Radix-2 CSD number through bit by bit conversion process.

C. EXISTING CSD MULTIPLICATION ALGORITHMS

CSD conversion is followed by CSD multiplications, which are of following types. They are Bit serial multiplication [12-14], Horner method based multiplication [1,6,7] and Pipeline CSD multiplication. In these first one is the direct method (i.e conventional method), second method is discussed below and third method discussed in the next section.

Horner based multiplication**:-** Horner method is used for Common Sub expression elimination. Here, this is applied for CSD multiplication so that required number of adders would be reduced. The procedure for doing this multiplication is given below with an example.

III. PIPELINE BASED MULTIPLICATION

The concept of pipelining [2],[3] is generally used to reduce the computation time. Here, this is used for the same purpose to pipeline CSD conversion processing step and CSD multiplication processing step, so that those two steps are overlapped to reduce overall time for the multiplication. This is clearly described in the Fig.2 below.

Here, B is a 4-bit multiplicand, M is a 4-bit multiplier. Binary number of

HORNER CSD MULTIPLICATION [1]:

For an example Consider the value of multiplicand as 165 and multiplier as 95

B=10100101(165)

M=01110011(95)

After CSD conversion M becomes C

C=1001’0101’

Where 1’ represents –1

This method begins looking for 1s from Left most bit and moving to the right side. Difference of positions of neighboring 1s is considered as weight as shown in the following expressions.

$B \text{ x } 2^3 – B = B_1$

$B_1 \text{ x } 2^2 + B = B_2$

$B_2 \text{ x } 2^2 – B = B_3$

Final Result = $B_3 \text{ x } 2^0$

Multiplier is converted into CSD number (C) by using Bit by bit CSD conversion. This type of multiplication is explained with an example as shown below.

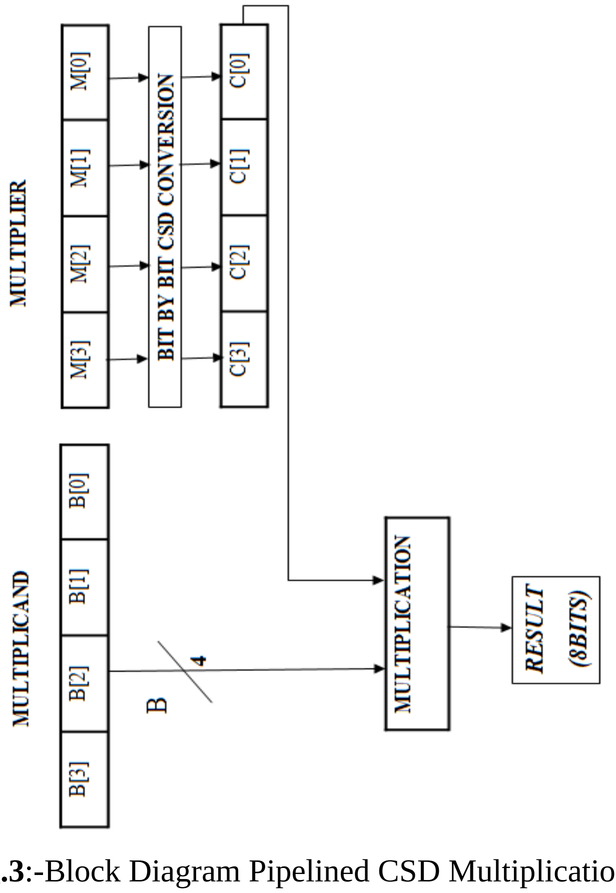

Fig.3:-Block Diagram Pipelined CSD Multiplication

PIPELINE CSD MULTIPLICATION:

For an example Consider the value of multiplicand as 12 and multiplier as 7

B=1100(12)

M=0111(7)

M is given to Bit by bit CSD conversion module as shown in Fig.2.

In the 1st step C[0] is produced as 1', then -1 is multiplied with Multiplicand, B. in the 2nd and 3rd steps as C[1] & C[2] are zeros, result is unchanged. In the 4th step Multiplicand shifted thrice (its position) and added to the previous result to get the final result.

Step1: C[0]= --1 => --B = B1

Step2: C[1]= 0 => No change

Step2: C[2]= 0 => No change

Step2: C[3]= 1 => B*2^3 + B1 = B2 (Final Result)

IV. RESULTS

In this section comparison between two CSD conversion techniques and comparison between existing and proposed CSD multiplication algorithms are presented.

Table1: Comparison between CSD conversion techniques.

Parameter	**Bit by bit CSD Conversion**	**Direct CSD Conversion**
Number of Slice LUTs	15 out of 63400	16 out of 63400
Number of bonded IOBs	24 out of 210	26 out of 210
Total delay	2.077 ns	2.088 ns
Power requirement	42.8 mw	42.8 mw

From the Table1 of comparison between Direct CSD conversion and Bit by Bit CSD conversion, it is shown that number of slice LUTs for direct CSD conversion technique is preferable over first technique. i.e. delay is lesser and number of slices and bonded IOBs are also lesser for Bit by bit CSD conversion technique than for Direct CSD conversion technique. Therefore Bit by Bit CSD conversion technique is applied in the following multiplication algorithms.

Table 2: Comparison between CSD multiplication algorithms with targeted device of Spartan6

Parameter	**Horner based multiplication[1]**	**Pipeline based multiplication**
Number of Slice LUTs	27916 out of 63400	233 out of 63400
Number of bonded IOBs	33 out of 210	33 out of 210
Number of BUFGs	1 out of 210	1 out of 210
Total real time for XST completion	756.0 sec	91.0 sec
Total CPU time for XST completion	755.31 sec	90.12 sec

Table 3: Comparison between CSD multiplication algorithms with targeted device of vertex-7

Parameter	**Horner based multiplication**	**Pipeline based multiplication**
Number of Slice LUTs	27924 out of 63400	213 out of 63400
Number of bonded IOBs	33 out of 210	33 out of 210
Number of BUFGs	1 out of 210	1 out of 210
Total real time for XST completion	709.0 sec	29.0 sec
Total real time for XST completion	708.31 sec	28.5 sec

Table2 and Table3 shows comparison between Horner based multiplication and pipeline based multiplication with targeted device of Spartan6 and Vertex-7 respectively.it shown that a lot of time of execution, requirement of number of slice LUTs are said to be saved in pipeline based multiplication in comparison with the Horner based multiplication, although number of bonded IOBs and number of BUFGs are equal for both of them in Spartan6 device. Same thing is applied even in Vertex-7. But from these comparisons, implementation on Vertex-7 device is said to be preferable over implementation on Spartan6 device. Because number of slice LUTs and total time for XST completion are greatly reduced.

V. CONCLUSION & FUTURE SCOPE

In the proposed method of radix-2 CSD multiplication Pipeline concept is applied and shown that it has saved execution time as well as the requirement of number of slice LUTs while maintaining same number of bonded IOBs and number of BUFGs in comparison with the Horner based method. Therefore DIP based systems or DSP based systems or NN based systems or any other related systems would be efficiently designed with this Pipeline based Radix-2 CSD multiplier with getting accelerated in speed and further miniature in size. Also, it has shown that implementation on Vertex-7 device is preferable over implementation on Spartan6 device.

In the future, this technique will be utilized for area and speed efficient design of fixed point multiplier or IEEE-754 single precision based floating point multiplier.

REFERENCES

[1] Keshab K.Parhi,“VLSI DIGITAL SIGNAL PROCESSING SYSTEMS: DESIGN AND IMPLEMENTATION”, John Wiley &sons Publishing Company Inc.,1999.

[2] M. Olivieri "Design of Synchronous and Asynchronous Variable-Latency Pipelined Multipliers", IEEE Trans. Very Large Scale Integr. Syst. vol. 9 no. 2 pp. 365-376 2001.

[3] H. Wold and A. M. Despain, “Pipeline and parallel-pipeline FFT processors for VLSI implementation,” *IEEE Trans. Comput.,* vol. C-33,

no. 5, pp. 414–426, May 1984.

[4] Gustavo A.Ruiz, Mercedes Granda. (2011, July.). efficient canonic signed digit recoding, S*cienceDirect Microelectronics journal* . [Online].42(2011) 1090-1097

Available: http://doi.org/10.1016/j.mejo.2011.06.006

[5] Rui Guo and Linda Sumners DeBrunner, “A novel fast canonical-sgned-digit conversion technique for multiplication,”in *Proc. ICASS-2011*,pp.1637-1640.

[6] Kripasagar Venkat, “Efficient multiplication and divison using MSP 430,” Texas Instruments, Dallas, Texas,SLAA329, Sep. 2006.

[7] Mayur N.Drukar, S.G.Bari. (2015, Mar.). Design of a parallel pipelined FFT architecture with reduced optimal delays, *IJMRD.*[Online].*2(3) ,* pp. 62–565.

[8] Manish Bansal, Sangeeta Nakhate and Ajay Somkuwar,“High performance pipelined 64X64-bit multiplier usingradix-32 modified booth algorithm and Wallace architecture,”in *Proc. ICCAIE,2010*,

pp.532–536.

[9] Ron S.Waters, Earl E.Swartzlander ”A Reduced Complexity Wallace tree multiplier reduction” in IEEE Transaction on computer, VOL.59, No.8,AUGUST,2010.

[10] V.G.Moshnyaga and K.Tamaru,”A comparative study of switching activity reduction techniques for design of low power multipliers,” IEEE International Sympoism on cicuits and systems,pp.1560-1563,1965.

[11] M.C.Wen, S.J.Wang and Y.M.Lin,”Low power parallel multiplier with column bypassing,” International Sympoism on cicuits and systems,pp.1638-1641,2005.

[12] Yunhua Wan, “Multiplier less CSD Techniques for high performance FPGA Implementation of Digital Filters,” Ph.D. dissertation, Dept. School of Elect. And Computer Eng., University of Okalahoma., Norman, Okalahoma,2007.

[13] Shoab Ahmed Khan,“ Multiplier-less Multiplication by Constants,” in Digital Design of signal processing systems: A practical approach, John Wiley & sons Publishing Company,2010,pp.253–299.

[14] Saroja.V Siddamal, R.M.Banakar and B.C.Jinaga (2008) Design of High-speed floating point multiplier. 4th International Sympoiusm on Electronic Design, Test And Applications.[Online].Available:http:// ieeexplore.ieee.org/document/4459558/

[15] Gustavo A.Ruiz, Mercedes Granda. (2011, July.). efficient canonic signed digit recoding, *ScienceDirect Microelectronics journal* . [Online].42(2011) 1090-1097

Available: http://doi.org/10.1016/j.mejo.2011.06.006*:*

[16] Rui Guo and Linda Sumners DeBrunner, “A novel fast canonical-sgned-digit conversion technique for multiplication,”in *Proc. ICASS-2011*,pp.1637-1640.