

Design and Implementation of Area Delay Efficient Booth Multiplier Based on CBL



G.Narender
M.Tech Student,
Department of ECE,
MVSR College of Engineering,
Hyderabad.



Sudhir Dakey
Assistant Professor,
Department of ECE,
MVSR College of Engineering,
Hyderabad.

ABSTRACT:

Multiplications and additions are most widely and more often used arithmetic computations performed in all digital signal processing applications. Addition is the basic operation for many digital applications. The aim is to develop area efficient, high speed and low power devices. Accurate operation of a digital system is mainly influenced by the performance of the adders. Multipliers are also very important component in digital systems. In this project, we designed a booth multiplier using CBL. The logic operations involved in conventional carry select adder (CSLA) and binary to excess-1 converter (BEC)-based CSLA are analyzed to study the data dependence and to identify redundant logic operations. We have eliminated all the redundant logic operations present in the conventional CSLA and proposed a new logic formulation for CSLA. In the proposed scheme, the carry select (CS) operation is scheduled before the calculation of final-sum, which is different from the conventional approach. Bit patterns of two anticipating carry words (corresponding to $C_{in}=0$ and 1) and fixed C_{in} bits are used for logic optimization of CS and generation units. An efficient CSLA design is obtained using optimized logic units. The proposed booth multiplier using CBL design involves significantly less area and delay than the recently proposed BEC-based CSLA.

KEYWORDS:

Booth Multiplier, Common Boolean logic Adder, Modified Carry Select Adder, Ripple Carry Adder, BEC.

1.INTRODUCTION:

Digital computer arithmetic is one of the main features of logic design with the aim of developing appropriate algorithms in order to optimise the utilization of the available hardware. The basic operations are multiplication, addition, division and subtraction. In this project, I am going to use the operation of additions in the operation of multiplication. The addition operations repeated and shifting results in the multiplication operations. Hardware can only perform a simple and limited set of operations. Arithmetic operations are based on a hierarchy of tasks (operations) that are built upon the simple tasks. In VLSI designs; area, speed and power are the mostly used measures for determining the efficiency and performance of the given architecture. Additions and Multiplications are most widely and more commonly used arithmetic operation performed in many digital signal processing applications.

All complex and simple digital multiplication is based on addition. An area efficient, fast and accurate operation of a digital system is greatly depends on the performance of the basic adders. Adders are very important component in digital logic design because of their wide use in these systems. Hence, to design a better architecture the basic adder blocks must have reduced delay time consumption and area efficient architectures. The demand is of DSP style systems for both less delay time and less area requirement for designing the systems. In the case of digital adders, the speed of addition is limited by the time required by the carry to propagate through the adder which is known as propagation delay time.

The sum for each bit in an adder is generated sequentially only after the previous bits have been summed and a carry is obtained to the next position. The carry select adder is used in many digital computational systems to reduce the problem of propagation delay. It can be done by independently generating multiple carries and then select a carry to generate the sum.

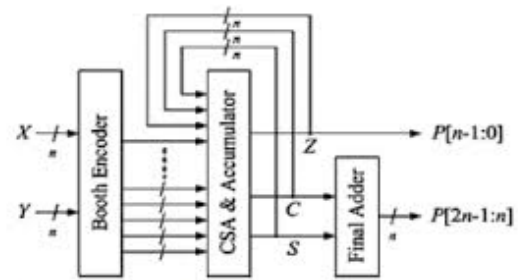
However, the CSLA is not efficient in the case of area because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input $C_{in}=0$ and $C_{in}=1$ separately, then the final sum and carry are selected by the multiplexer (mux). In the case of MCSLA the basic idea is to use Binary to Excess-1 Converter (BEC) instead of RCA with $C_{in} = 1$ in the regular CSLA to achieve lower area and power consumption

The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure. After obtaining the MCSLA; The adder in add and shift multiplier can be replaced. In VLSI design technique there are different types of multiplier structure are available. One of the basic multiplier is add and shift multiplier. This project deals with reduction of area, power requirement of add and shift multiplier without compromising to the speed of computation.

2. PROPOSED SYSTEM:

One of the most advanced types of MAC for general-purpose digital signal processing. It is an architecture in which accumulation has been combined with the common Boolean logic (CBL) tree that compresses partial products. In the architecture proposed in, the critical path was reduced by eliminating the adder for accumulation and decreasing the number of input bits in the final adder. While it has a better performance because of the reduced critical path compared to the previous MAC architectures, there is a need to improve the output rate due to the use of the final adder results for accumulation.

An architecture to merge the adder block to the accumulator register in the MAC operator was proposed in to provide the possibility of using two separate 2-bit adders instead of one -bit adder to accumulate the -bit MAC results.



Hardware architecture of the proposed MAC.

Fig: Block Diagram of Booth multiplier

III. PROPOSED ADDER DESIGN:

The proposed CSLA is based on the logic formulation given in figure and its structure is shown in Fig. It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG0 and CG1) corresponding to input-carry '0' and '1'. The HSG receives two n-bit operands (A and B) and generate half-sum word s_0 and half-carry word c_0 of width n bits each.

Both CG0 and CG1 receive s_0 and c_0 from the HSG unit and generate two n-bit full-carry words c_{01} and c_{11} corresponding to input-carry '0' and '1', respectively. The logic diagram of the HSG unit is shown in Fig. The logic circuits of CG0 and CG1 are optimized to take advantage of the fixed input-carry bits. The optimized designs of CG0 and CG1 are shown in Fig and (d), respectively.

The CS unit selects one final carry word from the two carry words available at its input line using the control signal c_{in} . It selects c_{01} when $c_{in} = 0$; otherwise, it selects c_{11} . The CS unit can be implemented using an n-bit 2-to-1 MUX. However, we find from the truth table of the CS unit that carry words c_{01} and c_{11} follow a specific bit pattern. If $c_{01}(i) = '1'$, then $c_{11}(i) = 1$, irrespective of $s_0(i)$ and $c_0(i)$, for $0 \leq i \leq n - 1$. This feature is used for logic optimization of the CS unit.

The optimized design of the CS unit is shown in Fig, which is composed of n AND-OR gates. The final carry word c is obtained from the CS unit. The MSB of c is sent to output as c_{out} , and $(n - 1)$ LSBs are XORed with $(n - 1)$ MSBs of half-sum (s_0) in the FSG to obtain $(n - 1)$ MSBs of final-sum (s). The LSB of s_0 is XORed with c_{in} to obtain the LSB of s .

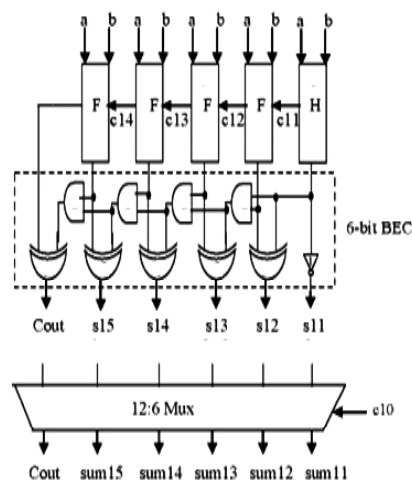
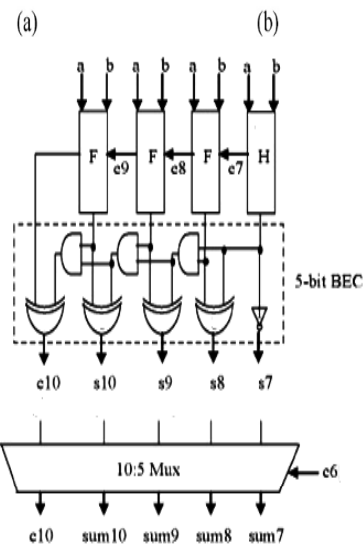
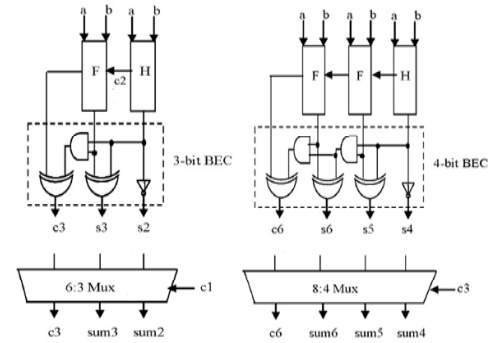
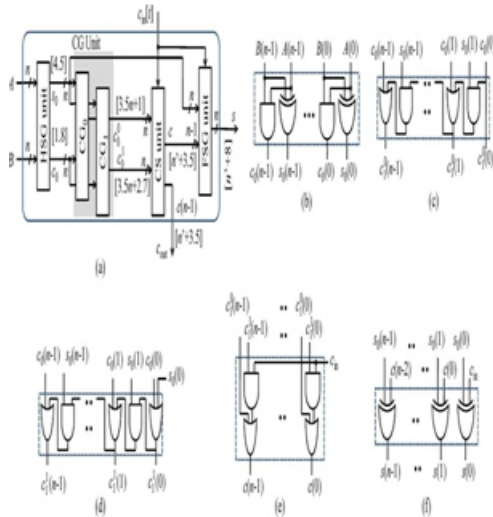


Fig.(a) Proposed CS adder design, where n is the input operand bit-width, (b) Gate-level design of the HSG. (c) Gate-level optimized design of (CGo) for input-carry = 0. (d) Gate-level optimized design of (CGi) for input-carry = 1.(e) Gate-level design of the CS unit. (f) Gate-level design of the final-sum generation (FSG) unit.

Modified 16 bit CSLA:

The architecture of the modified 16-b SQRT CSLA using Binary to Excess-1 converter for RCA with $C_{in} = 1$ to reduce the area and power is shown in Figure 5. We again split the structure into five groups which is shown Figure 4.

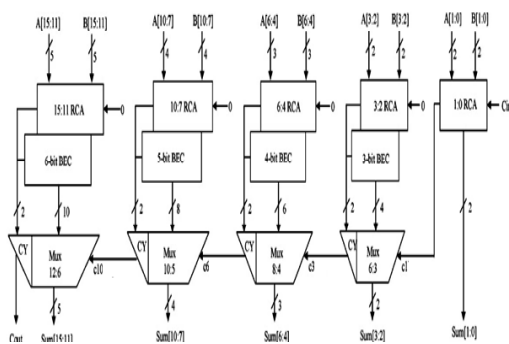


Figure 4: Modified 16-bit MCSL

Figure 5: Detailed connection: (a) group 2, (b) group 3,(c) group 4, and (d) group

The Blocks are ripple carry adder (RCA), binary to excess 1 converter (BEC) and Multiplexer. Each part is explained below

3.3. Block Diagram Details

3.3.1. BEC

As stated above in order to reduce the area and power consumption of the regular CSLA this project uses BEC instead of the RCA with $C_{in} = 1$. An $n+1$ -bit BEC is required to replace the n -bit RCA. The architecture and the function table of a 4-b BEC are shown in Figure 6 and Table 1, respectively.

Figure 9 illustrates the functionality of MCSLA. It gives the basic function of the CSLA is obtained by using the 4-bit BEC together with the mux. One of the inputs of the 8:4 mux is direct input (B_3, B_2, B_1 , and B_0) and other input of the mux is the output of BEC. This will result in two possible partial results in parallel. According to the control signal C_{in} the mux is used to select either the EC output or the direct inputs. The importance of the BEC logic is that this logic results in the large silicon area reduction when the CSLA with large number of bits are designed. The Boolean expressions of the 4-bit BEC is given as

$$\begin{aligned}
 X_0 &= \text{NOT}(B_0) \\
 X_1 &= B_1 \text{ XOR } B_0 \\
 X_2 &= B_2 \text{ XOR } (B_1 \text{ AND } B_0) \\
 X_3 &= B_3 \text{ XOR } (B_2 \text{ AND } B_1 \text{ AND } B_0)
 \end{aligned}$$

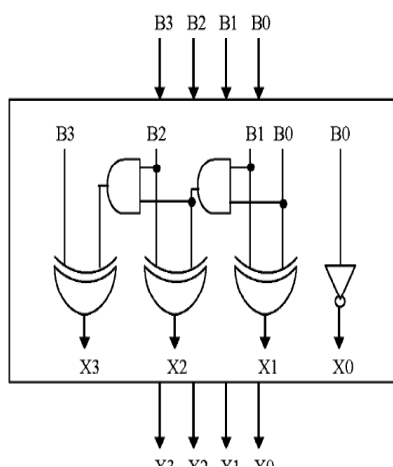


Figure 6: 4-b BEC.

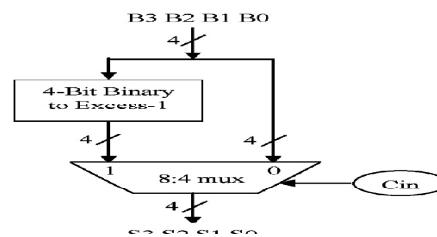


Figure 7: 4-b BEC with 8:4 MUX.

B[3:0]	X[3:0]
0000	0001
0001	0010
0010	0011
0011	0100
0100	0101
0101	0110
0110	0111
0111	1000
1000	1001
1001	1010
1010	1011
1011	1100
1100	1101
1101	1110
1110	1111
1111	0000

Table 1: Conversion table

3.3.2. RCA:

It is the well-known adder architecture. As shown in Figure 10 ripple carry adder is composed of cascaded full adders for 4-bit adder. RCA can be constructed by cascading full adder blocks in series. The carry out from one stage of full adder is fed to the carry-in of the next stage adder. 'n' full adders are required for an n-bit parallel adder. The dark line shows the carry flow from first full adder to the last.

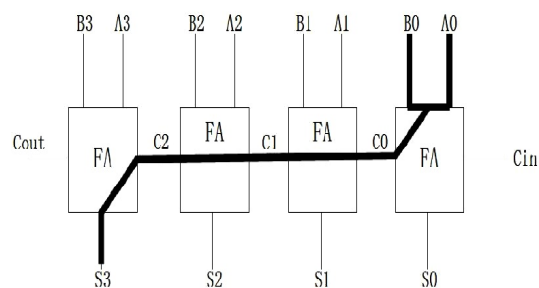


Figure 8: Ripple carry adder

When larger bit length numbers are used; RCA is not very efficient. Delay increases linearly with bit length. the carry-propagation chain will determine the latency of the whole circuit for a Ripple-Carry adder hence delay from Carry-in to Carry-out is more important than the delay from input to carry-out or carry-in to SUM. Figure 10. Shows ripple carry adder with carry flow.

3.3.3. Basic Adder Blocks:

An XOR gate is implemented by using AND, OR, and Inverter (AOI) as shown in Figure 11.

The gates between the dotted lines are performing the operations in parallel. That means both will execute same time. The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter, each having delay equal to 1 unit and area equal to 1 unit. By adding up the number of gates in the longest path of a logic block we will get the maximum delay. For each logic block the area evaluation is calculated by counting the total number of AND, OR, and NOT gates required.

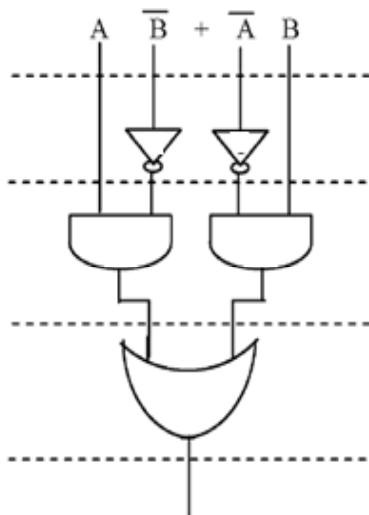


Figure 9: Adder.

4. SIMULATION RESULTS:

Codes for Multiplier and MCSLA are successfully verified by the simulation. Error conditions are intentionally made in the coding to check the complete functionality. Obtained utilization summary and simulated output is shown below Figure 12. This adder can be used for the construction of add and shift multiplier which have lowest area, high speed and minimum power consumption.

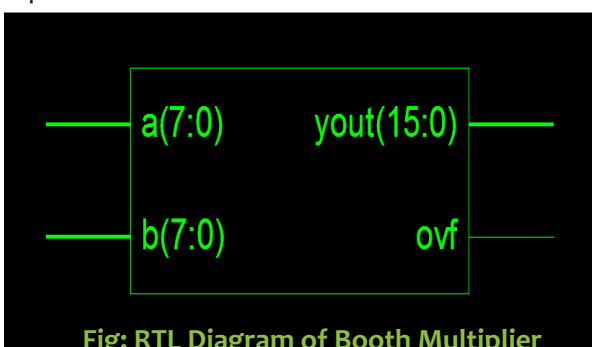


Fig: RTL Diagram of Booth Multiplier

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	128	4556	2%
Number of 4 input LUTs	223	9312	2%
Number of bonded I/Os	33	232	14%

Fig: Design summary of multiplier

Area and delay of multiplier:

	Time	Slices
MCSLA	30.012ns	114
CBL	33.377ns	128

Table 2: time and area:

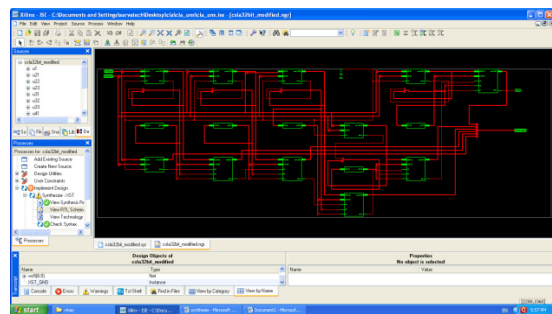


Figure 11: LUT optimization diagram

4.2. Simulated Output:

Figure 12: Simulated Output of MCSLA

The timing diagram displayed in Figure 13 shows one complete multiplication cycle of multiplier. This indicates from the Start signal to the Stop signal. The starting of computation is indicated by a start signal. Once the Stop signal is asserted at the end of the multiplication cycle; the result is obtained. From the figure, the Multiplier byte is 'A' and the Multiplicand byte is '96' so the expected result is 5DC.

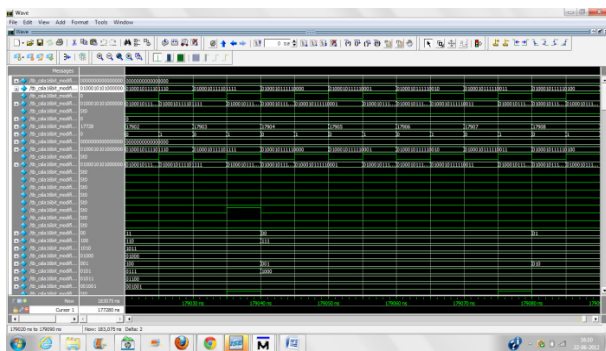


Figure 13: Simulated Output of Multiplier

5. Advantages:

- Cost effective compared to other proposed architectures
- High speed, Low power, Lower area
- Modified CSLA Can be used to implement Wallace tree Multiplier and Baug-Wooley Multiplier.

6. Applications:

- Data paths in Microprocessors.
- Digital Adders are the core block of DSP processors.
- Extensively used in processing units such as ALU.
- Forming dedicated integer and/or floating-point units.
- In Multiply-accumulate (MAC) structures.
- Digital Signal processing.
- High speed Integrated circuit

7. CONCLUSIONS:

Successfully achieved faster multiplier/adder structure using the Modified Carry Select Adder and CBL structure. With increasing word size, reduction of the delay increases; but the overhead of the area and power constraints decreases. The CBL adder is used to construct efficient Add and Shift Multiplier. CBL structure also can be used to make Wallace tree multiplier and Baugh-Wooley(BW) multiplier effectively. The proposed multipliers are energy efficient.

The proposed multiplier architecture can also be used to construct 32 bit, 64 bit and 128bit multiplier and significant speed can be achieved without much area or power constraints; that is, the 128-bit multiplier would be not only fast but also area, power, and energy efficient. The speed improvements are significant. Proposed techniques also improve the performance of multipliers. These design techniques can be implemented with all type of parallel multipliers of bit size higher than 16-b to achieve optimum performance without significant area and power constraints.

8. FUTURE WORK:

This 32 bit multiplier can be further extended to 64 bit multiplier and 128 bit multiplier using the proposed method for multiplication operation can be done as future work.

REFERENCES:

- [1] Ramkumar, B. and Harish M Kittur, (2012) "Low Power and Area Efficient Carry Select Adder", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp.1-5.
- [2] V. Vijayalakshmi, R. Seshadd, Dr. S. Ramakrishnan, (2013) Design and Implementation of 32 Bit Unsigned Multiplier Using CLAA and CSLA 978-1-4673-5301-IEEE.
- [3] He, Y. Chang, C. H. and Gu, J. (2005) "An Area Efficient 64-Bit Square Root Carry-Select Adder for Low Power Applications," in Proc. IEEE Int. Symp. Circuits Syst., Vol.4, pp. 4082-4085.
- [4] Padma Devi, Ashima Girdher and Balwinder Singh (1998) "Improved Carry Select Adder with Reduced Area and Low Power Consumption," International Journal of Computer Applications, Vol.3, No.4, pp. 14-18.
- [5] Akhilesh Tyagi, (1993) "A Reduced-Area Scheme for Carry-Select Adders," IEEE Transactions on Computers, Vol.42, No.10, pp.1163-1170.
- [6] Edison A.J and C.S. Manikandababu (2012.) "An Efficient CSLA Architecture for VLSI Hardware Implementation" International Journal for Mechanical and Industrial Engineering, Vol. 2. Issue 5.

- [7] P.Sreenivasulu, K.SrinivasaRao, Malla Reddy, and A.VinayBabu(2012) "Energy sand Area efficient Carry Select Adder on a reconFigureurable Hardware" International Journal of Engineering Research and Applications, Vol. 2, Issue 2, pp.436-440.
- [8] Sarabdeep Singh and Dilip Kumar, (2011) "Design of Area and Power efficient Modified Carry Select Adder" International Journal of Computer Applications (0975 – 8887) Volume 33– No.3.
- [9] S. Brown and Z. Vranesic, (2005) Fundamentals of Digital Logic with VHDL Design, 2nd ed., McGraw-Hill Higher Education, USA,. ISBN:0072499389.
- [10] P. C. H. Meier, R. A. Rutenbar and L. R. Carley(1996), "Exploring Multiplier Architecture and Layout for low Power", CIC'96.
- [11] HasanKrad and AwsYousi(2010)"Design and Implementation of a Fast Unsigned 32-bit Multiplier Using VHDL".
- [12] Sreehari Veeramachaneni, Kirthi M, Krishna Lingamneni, Avinash Sreekanth Reddy Puppala M.B.Srinivas(2007), 'Novel Architectures for High-Speed and Low-Power 3-2, 4:2 and 5:2 Compressors', 20th International Conference on VLSI Design, Pp: 324-329.
- [13] S. F. Hsiao, M. R. Jiang, and J. S. Yeh, (1998.) 'Design of high speed low-power 3:2 counter and 4:2 compressor for fast multipliers', Electron. Lett, vol. 34, no. 4, Pp. 341-343.
- [14] K. Prasad and K. K. Parhi, (2001) 'Low-power 4:2 and 5:2 compressors,' in Proc. of the 35th Asilomar Conf. on Signals, Systems and Computers, vol. 1, , Pp.129-133.
- [15] Massimo Alioto and Gaetano (2002), 'Analysis and Comparison on Full Adder Block in Submicron Technology', IEEE Transaction Very Large Scale Integration (VLSI) Systems, Vol 10, No. 6, Pp: 806-823.
- [16] Anantha P. Chandrakasan, Samuel Sheng and Robert W. Brodersen (1992), 'Low-Power CMOS Digital Design', IEEE Journal of Solid State Circuits, Vol.27, No. 4.
- [17] P. S. Mohanty, (2009.) "Design and Implementation of Faster and Low Power Multipliers", Bachelor Thesis. National Institute of Technology, Rourkela.