Estimation of Speed and Area of 32-Bit Synchronous Multiplier Designed Using Brent-Kung Prefix Adder

Rakesh Kumar

***Abstract*— Multiplier is the most important element of the digital signal processing such as filtering and convolution and hence their area and speed are of prime concern.Here, in this paper we have designed and implemented a synchronous Baugh-wooley Multiplier for 32-bits multiplication. Designing and verification is done through VHDL on Xilinx8.1. In this paper we tried to explain the step by step process that was adopted for Signed-Unsigned BWMultiplier. By comparing a few multipliers we get the best solution to optimize the speed and area.**

Also, two different approaches for 32 bit multiplication proposed and after, in implementation we could see the differences in certain parameters. The array structure of Booth's Multiplier and Signed-Unsigned Baugh-Wooley Multiplier is obtained from RTL synthesis are shown. Different parameters like power, CPU usage, memory usage and area etc. have been compared.

***Index Terms*— VHDL, Multipliers, Baugh-Wooley, filtering, Booth multiplier**

I. INTRODUCTION

As known, multipliers are most important integral function in arithmetic operations[1]. In some applications of digital signal processing (DSP) like convolution and in FFT some of the computational arithmetic function are applied regularly (e.g. Multiply and Accumulate, inner products) [2] hence their power dissipation and speed are of prime concern.However speed and area are two important constraints. So improving speed results always in large areas[5]. Many research efforts have been devoted to reducing the power dissipation and reducing area of different multipliers. Generation of the partial product is the largest contribution in total power consumption. So in our thesis we have tried to present a fixed width BW multiplier using full-width BW multiplier which computes the 2n output as a weighted sum of partial-products using Brent-Kung adder. BW algorithm is an efficient way to handle the sign bit. BW multiplier uses full adders, half adders and can be implemented by other parallel prefix adder. The main aim of this project is to reduce the area and power consumed by the fixed-width signed multiplier which uses both half adders and full adders for implementation.

Manuscript received April 20, 2014.

Rakesh Kumar, is pursued B.Tech. From Rajasthan Technical University Kota in 2011 and he is pursuing his M. Tech. in EMBEDDED SYSTEM from JNU JAIPUR.

II. PREVIOUS WORK

A. *Booth multiplier using CSA*

a) Booth encoding

Booth's encoding or Booth's multiplication algorithm [3] is a multiplication process which can multiply two signed binary numbers in a 2's complement notation. Booth's algorithm has the ability to perform fewer additions and subtractions in comparison to other multiplication algorithm. This encoding process is used to minimize the no of partial products in a multiplication process and based upon the relation

$2n = 2n+1 – 2n$

Booth's algorithm checks consecutive bits of the N-bit multiplier y in signed 2's complement representation, it includes an implicit bit below the least significant bit, $y\text{-}1 = 0$. For every bit y_i, as i run from 0 to N-1 [3], bits $y_i$and $y_i\text{-}1$ are considered. When these two bits are equal, the product accumulator P stays unchanged. Where $y_i = 0$ and $y_i\text{-}1 = 1$, the multiplicand times 2i is added to P; and where $y_i\text{-}1 = 0$and $y_i = 1$, the multiplicand times 2igets subtracted from P. The final value of P will be the signed product.

b) Modified Booth algorithm

The Booth algorithm was invented by A. D. Booth forms the base of Signed numbermultiplication algorithms that are simple to implement at the hardware level[4] and thathave the potential to speed up signed multiplication considerably. Booth's algorithm isbased on recoding of multiplier y, to a recoded value z and leaving the multiplicandx unchanged. In Booth recoding, each digit of the multiplier can assume positive aswell as negative and zero values. There is a special notation, called signed digitencoding. In signed digit encoding +1 and 0 are expressed as 1and 0, but -1 is expressed as 1.

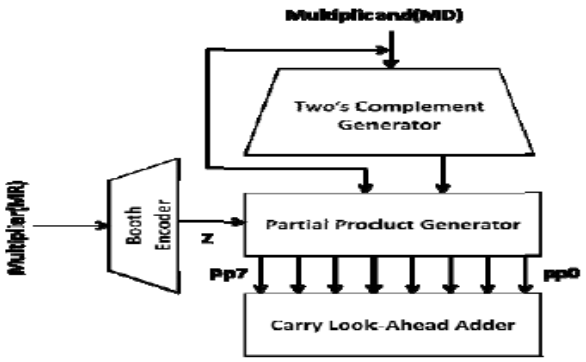

Fig.1. Architecture of booth multiplier [4]

The value of a 2's complement integer was defined by equation 1.

$$y = -y_{m-1}2^{m-1} + \sum_{i=0}^{m-2} y_i 2^i$$

This equation shows that in order to get the value of a signed 2's complementnumber, multiply the m – ith digit by -2`-1, and multiply each remaining digit i by +2g.
For example (-7), which is 1001 in 2's complement notation, would be, in SDnotation, 1001 = -8 + 0 + 0 + 1 = -7.
For implementing booth algorithm important step is booth recoding. By boothrecoding we can replace string of 1s by 0s. For example the value of strings of five 1s,
11111 = 2^5 - 1 = 100001 = 32 – 1 = 31. Hence if this number could to be used as themultiplier in a multiplication, we have to replace five additions by one subtraction and oneaddition[4].
The Booth recoding process then follows as:
1. Moving from LSB to MSB, replace each 0 digit of the original number with a 0in the recoded number until a 1 is encountered.
2. When a 1 is encountered, place a 1 at that position in the recoded number, andskip over any succeeding 1's until a 0 is encountered.
3. Replace that 0 with a 1 and continue.
This algorithm is shown in tabular form in Table 1, considering pairs ofnumbers, y_i,$y_{i-1}$and the recoded digit, z_i, as shown in Table 1.

Table1. Booth algorithm

y_i	y_{i-1}	z_{i-i}	Multiplier value	situation
0	**0**	**0**	**0**	**String of 0s**
0	**1**	**1**	**+1**	**End of sting of 1s**
1	**0**	**1**	**-1**	**Begin string of 1s**
1	**1**	**0**	**0**	**Sting of 1s**

c) Booth Algorithm

1. Add 0 to right of LSB of multiplier and look at rightmost of multiplier to make pairing of 2 bits from right to left and mark corresponding multiplier value as shown in table.
2. 00 or 11 : do nothing
3. 01: mark the end of a string of 1s and add multiplicand to partial product.
4. 10: Marks the beginning of a string of 1s subtract multiplicand from partial product.

B. Asynchronous 32 bit BW multiplier using full adder & half adder

Asynchronous 32 bit BW multiplieris the best known algorithm for signed multiplication because it maximizes the regularity of the multiplier and allows all the partial products to have positive sign bits.Baugh–Wooley technique was designed for thedirect multipliers for 2's compliment numbers.When multiplying 2's compliment numbers directly,each of the partial products to be added is a signed numbers. Hence each partial product has to be sign extendedto the width of the final product in order to form a correct sum by full adder and half adder tree.

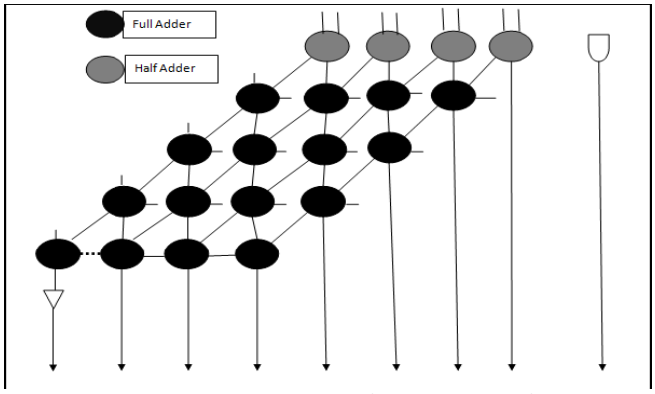

Fig.2. Baugh-Wooley multiplier using full adders and half adders [2]

III. PROPOSED WORK

A. Brent-Kung prefix adder

The Brent-Kung adder is a parallel prefix adder. Parallel prefix adders are special class of adders that are based on the use of generate and propagate signals. Simpler Brent-Kung adders was been proposed to solve the disadvantages of Kogge-Stone adders. The wiring complexity and cost is greatly reduced. But the logic depth of BK adders [8] increases to 2log (2n-1), so the speed is lower. The block diagram of 4-bit Brent-Kung adder is shown in Fig. 3.

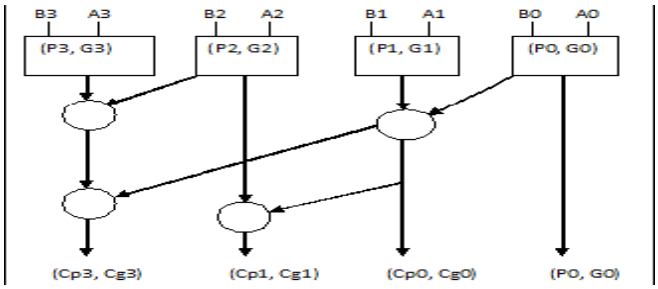

Fig.3.4-bit Brent-Kung adder [6]

This algorithm is based on a dconquer and divide approach, in which inputs are combined pair-wise toobtain the sequence of length k/2 and then even-indexed prefix are computed by the odd-indexed. The mainadvantage of this architecture is its less processing nodes and parallel generation of carries and less layout area.

B. Synchronous BW multiplier using BK adder

BW Multiplication involves two basic operations: the generation of the partial product and their accumulation.Therefore, there are possible ways to speed up the multiplication: reduces the complexity, and as a result reduces the time needed to accumulate the partial products.
Brent-Kung adder (BK adder) is an advanced design prefix adder, which is a very good balance between area and power cost and also it will present better performance. BK adder has a complex carry and inverse carry tree. A tree can bedivided into 2 types that are a tree and an inverse tree. Upper tree is based on periodic power of 2.

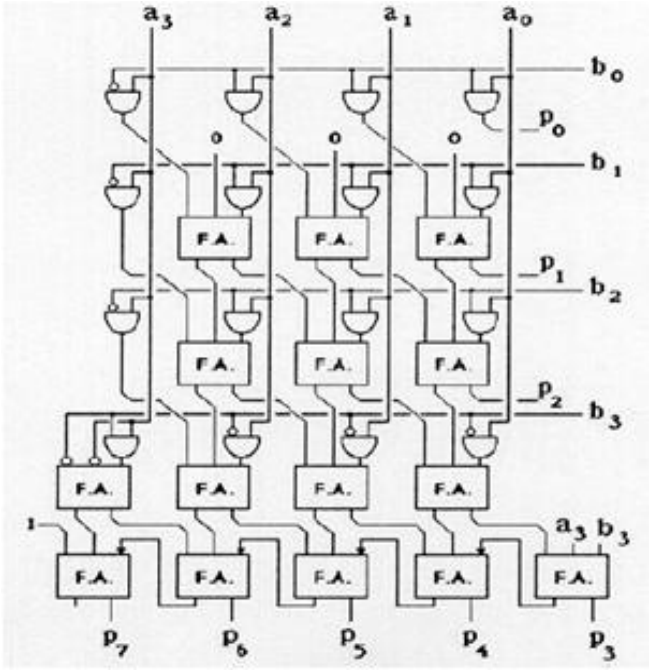

Fig.4. Block diagram of a 4*4 Baugh-Wooley multiplier [9]

Fig.4. Block diagram of a 4*4 Baugh-Wooley multiplier [9]

Baugh-Wooley Multiplier is used for both unsigned and signed number multiplication[2]. Signed Number operands which are represented in 2's complemented form. All the Partial Products are adjusted such that negative sign move to final step, which in turn maximize the integrity of the multiplication array. Baugh-Wooley Multiplier works on signed operands with 2's complement representation to make sure that the signs of all partial products are positive.

IV. RESULTS

We have done the coding in VHDL of different multipliers discussed above. The VHDL codes of different multipliers are synthesized on Xilinx ISE 8.1 and simulated using Xilinx ISE simulator. The device utilization and timing summary obtained from synthesis report is used to compare the different multipliers. As we know that LUTs is proportional to the area occupied by the multipliers on VLSI chip and path delay is inversely proportional to the speed of the multiplier. These coding are also written in VHDL language and simulate it to get the RTL circuit of each system. Also get the lookup table, where we get the exact no of inputs, outputs and no of slices requirement etc for the system. Xilinx Power Estimator is used to determine&analyze the power consumption of the system. These results of multipliers are given below.

Table2. Synthesis report of booth multiplier using CSA

Device utilization summary (estimated values)			
Logic utilization	Used	Available	Utilization
Number of Slices	1282	10752	11%
Number of 4 input LUTs	2319	21504	10%
Number of bonded IOBs	129	448	28%

Maximum combination path delay: 28.485ns

Table3. Synthesis report of asynchronous BW multiplier

Device utilization summary (estimated values)			
Logic utilization	Used	Available	Utilization
Number of Slices	1037	10752	9%
Number of 4 input LUTs	2041	21504	9%
Number of bonded IOBs	128	448	28%

Maximum combination path delay: 48.09 ns

Table4. Synthesis report of synchronous BW multiplier using BK adder

Device utilization summary (estimated values)			
Logic utilization	Used	Available	Utilization
Number of Slices	2254	10752	20%
Number of 4 input LUTs	2130	21504	9%
Number of bonded IOBs	130	448	29%

Maximum combination path delay: 6.496ns

Table4. Comparison of different multipliers

Logic utilization	Booth multiplier	Asynchronous BW multiplier	Synchronous BW multiplier
Number of slices	1282	1037	2254
Number of 4 input LUTs	2319	2041	2130
Number of bonded IOBs	129	128	130
Number of GCLKs	1	1	1
Maximum combination path delay	28.485	48.09	6.496

V. CONCLUSION

In this paper, a 32 bit synchronous multiplier proposed in which partial products are generated using BK adder. The device utilization summary obtained from synthesis report, shows that synchronous BW has less number of LUTs as compare to other multiplier which shows BW has less area in comparison to the other multiplier. By simulating the proposed synchronous BW multiplier we found that Maximum combinational path delay in this multiplier is 6.496ns, which is very less as compare to other multipliers. This technique performs the multiplication of 32 bit. When we compare the path delay and LUTs of synchronous BW multiplier with other multipliers, we found that synchronous BW multiplier using BK adder is better than other multipliers in terms of speed and area. So by using synchronous BW multiplier we can achieve the fast and efficient multiplication.

REFRENCES

[1] Anand Kumar., "*Fundamentals of Digital Circuits*", *Prentice Hall of India*, 2008.
[2] Manish Chaudhary, Mandeep Singh Narula., " *FPGA Implementation of Booth's and Baugh- wooley multiplier Using Verilog*" in International

Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-3, Issue-1, June 2013

[3] A. D. Booth "*A signed binary multiplication technique*," Quart. J. Mech. Appl. Math.,
vol.4

[4] Sukhmeetkaur, suman and manpreetsignh manna., "*Implementation of Modified Booth Algorithm (Radix 4) andits Comparison with Booth Algorithm (Radix-2)*" in Advance in Electronic and Electric Engineering.
ISSN 2231-1297, Volume 3, Number 6 (2013), pp. 683-690

[5] Vikram Singh, Manish kumarjain., " *Estimation of Speed and Area of High Speed Multiplier Designed using Booth-Wallace Unit Add Method*" in International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July 2013

[6] C. S. Wallace., "*A Suggestion for a Fast Multiplier*", *in IEEE Trans. on Electronic Computers*, vol. 13, pp. 14-17, 1964.

[7]. Morris Mano,,Digital Design, Third edition', "*Prentice Hall of India",* 2000.

[8] AdilakshmiSiliveru, M.Bharathi., "*Design of Kogge-Stone and Brent-Kung adders using Degenerate Pass Transistor Logic*" in International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-1, Issue-4, February 2013

[9] IndrayaniPatle, AkanshaBhargav, PrashantWanjari., "*Implementation of Baugh-Wooley Multiplier Based on Soft-core Processor*" in IOSR Journal of Engineering (IOSRJEN) e-ISSN: 2250-3021, p-ISSN: 2278-8719 Vol. 3, Issue 10 (October. 2013), ||V3|| PP 01-07

Author Profile:

Rakesh Kumar is pursued B.Tech. From Rajasthan Technical University Kota in 2011 and he is pursuing his M. Tech. in EMBEDDED SYSTEM from JNU JAIPUR. His interested research fields are Low power VLSI and Embedded System Design, Communication and Signal Processing Systems.