VLSI Based Energy-efficient Multipliers with double LSB operands

Lanka. Keerthi[1], Dr.M.Satyanarayana[2]

[1]PG scholar, Dept. of ECE, [2] Prof, ECE Dept
MVGR College of Engineering, Vizianagaram, A.P, India-535005.

Abstract: *In this paper, various multipliers are implementing for low power requirement and high speed. The efficiency can be implementing number mathematical approaches. The main focus is energy-efficient scheme for multiplying 2'scomplementbinary numbers with two least significant bits (LSBs).The implementation in hardware is done by using FPGA of the proposed logic in circuit, by using technology at 45-nm technology the area of the processor is 1.89mm2. The processor supports 972 MS/s at 250 MHz with a power consumption of 29 mW and a signal-to-quantization-noise ratio of 52.16 dB.*

Key words: *LSB, DLSB, FPGA*

1 Introduction

Performance is recognized as a 1 of the critical parameter in digital system design field, especially in digital signal processing (DSP) applications. In digital signal processing systems Fast multipliers are essential parts. The speed of multiply operation is of great importance in DSP also as within the overall purpose processors today, especially since the media processing took off [1].previously multiplication is generally implemented via a sequence of Subtraction, addition and shift operations.Multiplication operation is that the series of repeated additions, number to be added is that the multiplicand and therefore the number of times that it's added is that the multiplier, and the result is the product. Each step of addition generates a partial product. In most computers, the operand usually contains an equivalent number of bits. When the operands are interpreted as integers, the merchandise is usually twice the length is decided by the performance of the multiplier because the multiplier is usually the slowest clement in the system. In digit, serial multipliers have single digits consisting of several bits are operated on. In both speed and area, these multipliers have moderate performance. However, existing digit serial multipliers are suffering from complicated switching systems or irregularities in design. The parallel Multipliers provide high performance compared to serial multipliers and avoiding irregularities [8]. In this paper, evaluating performance of three different parallel multipliers in terms of power, delay and area and determine which one is providing high performance of operands in order to preserve the information content[3],[4]. This repeated method of addition that is suggested by the definition of arithmetic is slow that it's nearly always replaced by an algorithm that creates the use of positional representation. It is easy to decompose multipliers into two parts. The first part is devoted to the generation of partial products and therefore the other collects and adds them. The basic multiplication principle is twofold, evaluation of partial products and accumulation of the shifted

partial products and the second one collects and adds them [2]. The basic multiplication principle is two-folded, accumulation of the shifted partial products and evaluation of partial products. It is performed by the successive Addition's of the columns of the shifted bit of the 'multiplicand.[6],[7]. The delayed, gated instance of the multiplicand must all be of the shifted partial product matrix in the same column. They are then added to form the product bit for the particular form .Multiplication is there for multi operand operation. To extend the multiplication to both unsigned and signed numbers a convenient. Number system is the representation of numbers in 2's complement format. In arithmetic function Multiplication is an important fundamental function.

2 Literature survey

Multipliers play an important role in today's digital signal processing and various other applications [9]. With advances in technology, many researchers are trying to design multipliers which offer either of the following design targets – high speed, power consumption should be low, regularity of layout and hence area should be less or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation[11].

2.1.Characteristics of multiplier:

An efficient multiplier should have following characteristics:

Speed: Multiplier should perform operation at high speed.

Area: multiplier should occupy less number of slices and LUTs.

Power: Multiplier should consume less power[13]. The definition, takes O(N2) arithmetical operations, while an FFT can compute the same result in only O(N log N) operations[10]. The difference in speed can be substantial, especially for long data sets. where N will be in the thousands or millions, in practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to N / log (N)[5].

1. Partial product generation.
2. Partial product reduction.
3. Final addition.

For the multiplication of an *n*-bit multiplicand with an *m* partial products, *m*-bit multiplier are generated and product formed is *n* + *m* bits long. Here there are four different types of multipliers which are

1. Booth multiplier.
2. Combinational multiplier.
3. Wallace tree multiplier.
4. Array multiplier.
5. Sequential multiplier.

3.Booth multiplier:-

Booth multiplication algorithm gives a procedure for multiplying binary integers in signed -2's complement representation[12]. Following steps are used for implementing the booth algorithm:-Let X and Y are two binary numbers and having m and n numbers of bits(m and n are equal) respectively.

Step 1 Making booth table: In booth table we will take four columns one column for multiplier second for previous first LSB of multiplier and other two (U and V) for partial product accumulator (P).

1. From two numbers, choose multiplier (X) and multiplicand (Y).
2. Take 2's complement of multiplicand (Y).Load X value in the table.
3. Load 0 for X-1 value.
4. Load 0 in U and V which will have product of X & Y at the end of the

operation.

5. Make n rows for each cycle because we are multiplying m and n bits numbers .

Step2 Booth algorithm: Booth algorithm requires examination of the multiplier bits, and shifting of the partial product(P). Prior to the shifting, the multiplicand may be added to P, subtracted from the P, or left unchanged according to the following rules:

1. Xi Xi-1

0 0 Shift only

1 1 Shift only

0 1 Add Y to U and shift

1 0 Minus Y from U and shift

1.Take U & V together and shift arithmetic right shiftwhich preserves the sign bit of 2's complement number.So, positive numbers and negative numbers remainspositive and negative respectively.

2.Circularly right shift X because this will prevent usfrom using two registers for the X value.

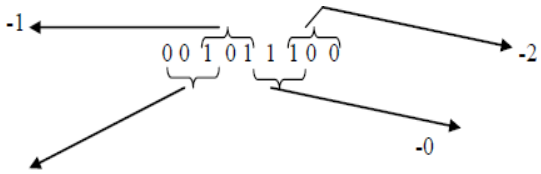
3.1. Modified booth multiplier:

It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product[7]. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better

Example: Multiplier is equal to 0 1 0 1 1 10 then a 0 is placed to the right most bit which gives 0 1 0 1 1 10 0 the 3 digits are selected at a time with overlapping leFT most bit as follows

$$Z_i = -2xi + 1 + xi + xi - 1 \quad (1)$$

TABLE-1



B	Zn	Partial Product
000	0	0
001	1	1×Multiplicand
010	1	1×Multiplicand
011	2	2×Multiplicand
100	-2	-2×Multiplicand
101	-1	-1×Multiplicand
110	-1	-1×Multiplicand
111	0	0

Fig1 Block Diagram of Modified Booth Algorithm

4. DLSB multipliers

Let $A^+ = \langle a_{n-1} a_{n-2} \cdots a_0 \rangle_{2's} + a_{0+}$ and $B+ = \langle b_n - 1 b_n - 2 \cdots b_0 \rangle_{2's} + b_0$ (2)

$n - 2a_i 2^i = -\langle a_n - 1 a_n - 2 \cdots a_0 \rangle_{2's}$ (3)

define A^+ using the expression $a_i' = a_i \oplus a_0$ +. Thus, we get be two n-bit 2's-complement binary numbers with DLSB. Their product is defined as follows:

$$A^+ = (-1)^{a0} + {}_{-a'n-12}n - 1 + n \sum - 2_{a'i} 2^i \quad (4)$$

$$P = A^+ \cdot B^+ = (\langle a_{n-1} a_{n-2} \cdots a_0 \rangle_{2's} + a_{0+}) \cdot B + (5)$$

$i = 0$

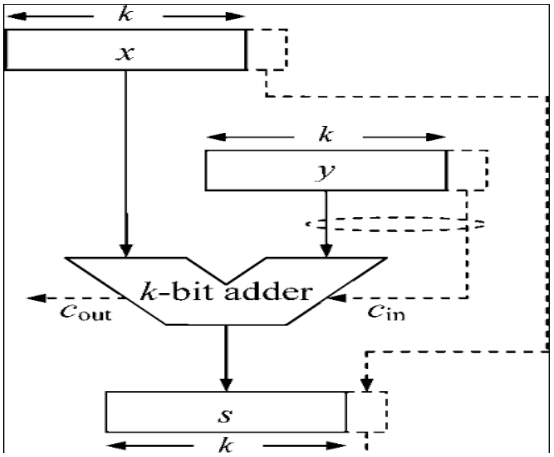


Fig2.Addition of two DLSB numbers.

4.1. Proposed DLSB implementation

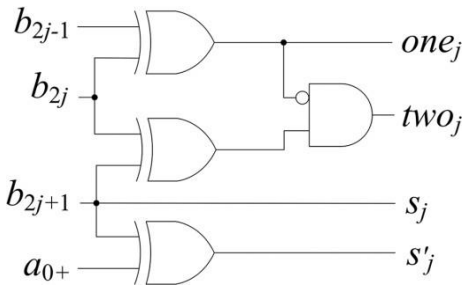


Fig3.Proposed DLSB Modified Booth encoder

$a_{0+}=0$, we get Input MB digit Output b_{2j+1} $b2j$ b_{2j-1} bMB $_{jsign}=s_j$ × 2 = two_j × 1 = one_j 0000000 (5)

$A^+ = \langle a_{n-1}an-2\cdots a0\rangle 2's$ (7) ii) $a_{0+}=1$, and using the expression $ai = -\bar{a}i+1$, we get (6)
0 0 1 1 0 0 **1** 0 1 0 1 0 0 **1** 0 1 1 2 0 **1 0**

$A^+ = \langle a_{n-1}an-2\cdots a0\rangle 2's + 1 = -a_{n-1}2n-1+n\sum_{i=0}-2_{ai}2i+1$ (7)
1 0 0 −2 **1 1 0** 1 0 1 −1 **1** 0 **1**
$= -(-\bar{a}n-1+1)2^{n-1}+n\sum-2_{(-\bar{a}i}+1)2^i+$
1 1 1 0 −1 **1** 0 **1**
1 1 1 **0 1 0 0**

$= -(-\bar{a}n-12^{n-1}+2^{n-1})-n\sum-2_{\bar{a}i2}i+n\sum-2_2i+1$ (8)

4.2. Power vs. Energy Calculations

PDP is the average energy consumed per switching event/Takes into account that one can trade
delay2 Product (EDDP) = EDP * t (10)
Power ~ ½ CV^2Af, Performance ~ f (and V) (11)

Power ~ CV^2f ~ V^3 (fixed icroarch/design) (12)

Performance ~ f ~ V (fixedmicroarch/design) (13)

Analyzed the area occupied, power consumed and the time delay consumed by different multipliers and found out best among them. After comparing all we came to a conclusion that Modified booth multiplier is best suited for high speed and low power applications

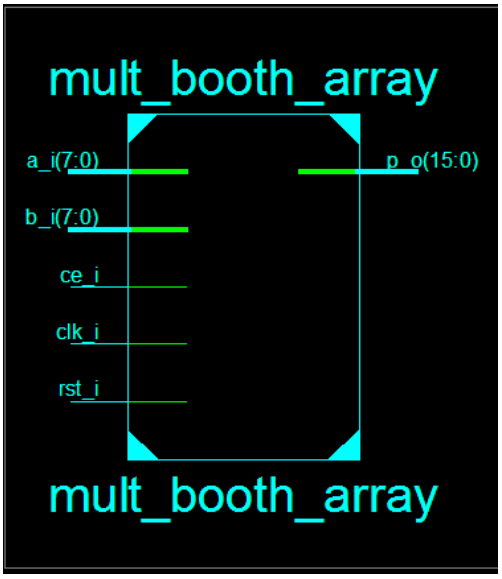


Fig4.Booth Multiplier synthesized RTL Schematic

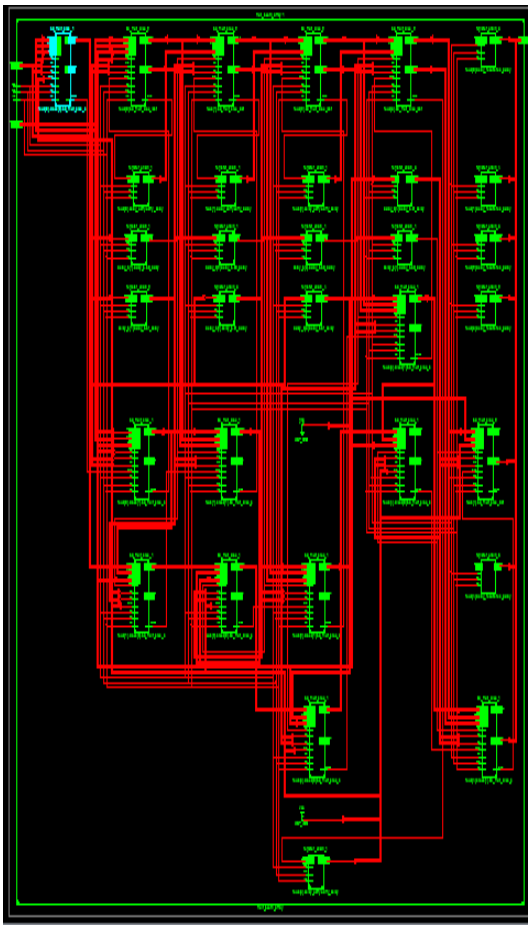


Fig: 5Internal; structure of Booth Mulitiplier

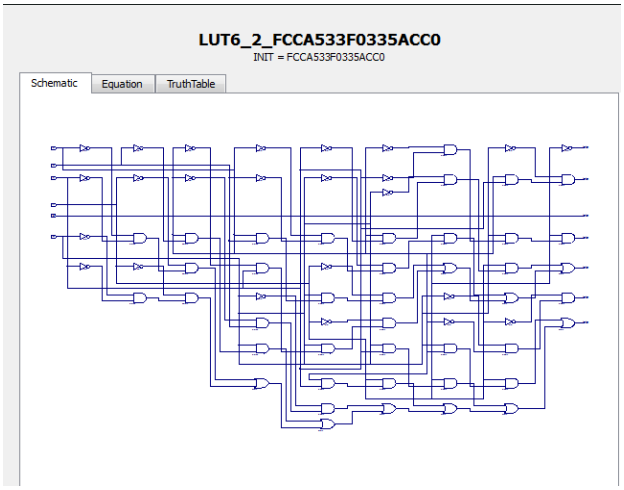


Fig:6 Internal structure of Look up table circuit

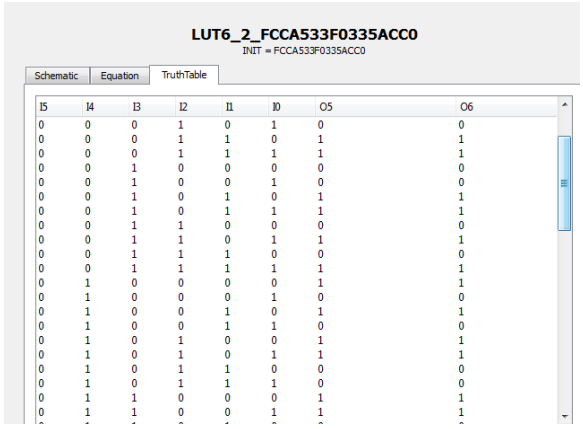


Fig:7 Internal structure of Look up table circuit output

```
--------------------------------------------------------------------
Release 9.2i - XPower SoftwareVersion:J.36
Copyright (c) 1995-2007 Xilinx, Inc.  All rights reserved.
Design:       E:\personal\PROJECTS\ssit\New folder\xzz\Boot.ncd
Preferences:  Boot.pcf
Part:         xqvr300cb228-4
Data version: PRODUCTION,v1.0,05-28-03

Power summary:                          I(mA)     P(mW)
--------------------------------------------------------------------
Total estimated power consumption:                  29
                                          ---
                          Vccint 2.50V:       9       23
                          Vcco33 3.30V:       2        7
                                          ---
                               Inputs:        0        0
                                Logic:        0        0
                              Outputs:
                                 Vcco33       0        0
                              Signals:        0        0
                                          ---
                  Quiescent Vccint  2.50V:       9       23
                  Quiescent Vcco33  3.30V:       2        7

Thermal summary:
--------------------------------------------------------------------
   Estimated junction temperature:                  26C
                     Ambient temp:  25C
```

Fig: 8.Power Analysisreport

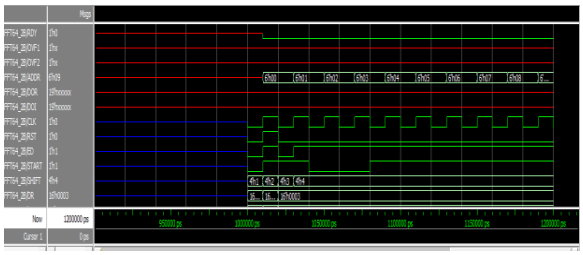


Fig:9 Booth Multiplier simulation output

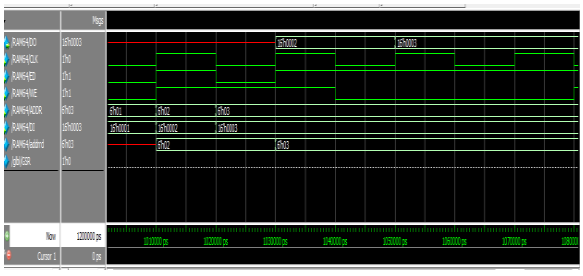


Fig 10. Modified Booth Multiplier simulation output

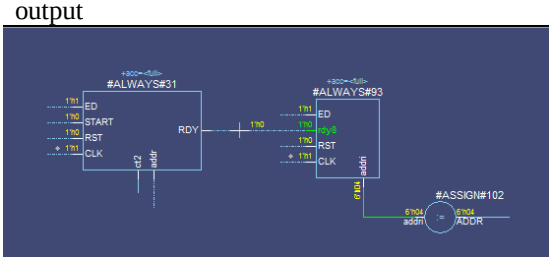


Fig:11 Multiploerunit Dataflow output

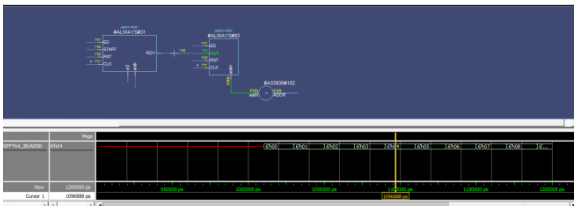
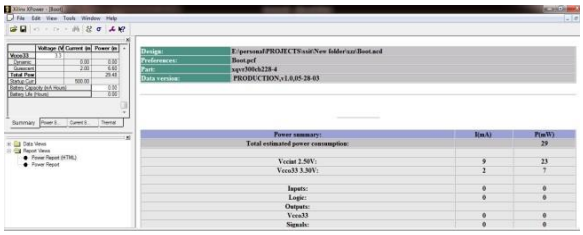


Fig:12 Modified Booth Dataflow output

Table-2 Logic utilization % Table

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	150	4800	3%
Number of Slice LUTs	64	2400	2%
Number of fully used LUT-FF pairs	51	163	31%
Number of bonded IOBs	35	102	34%
Number of BUFG/BUFGCTRLs	1	16	6%



5.Recommendations & Future Work

In this project work we propose a high speed low-power multiplier and accumulator adopting the new modified Booth implementing approach. On a modified Booth algorithm works on encoder side is controlled 2s complement of given number converted to equvallent decimal value or in binary bits The modified booth algorithm works for reduce the logic of 2's complemented . The booth adder will avoid the unwanted addition and thus minimize the switching power dissipation. The booth implementation with AND gates have an extremely high flexibility on adjusting the data asserting time., while changing the original k with the one designed in Verilog allowed for the same results.

CONCLUSION

We analyzed the area occupied, power consumed and the time delay consumed by different multipliers and found out best among them. Implement and r comparing came we came a conclusion that the best application multiplier is Modified booth multiplier is good suited device for high speed and low power applications.

FUTURE WORK

A lot of scope for future research directions are possible for reduce the area and power and time delay consumption. My possible directions for convolution multipliers by using Modified booth techniques while implementation high speed communication devices or image enhance devices.. Further reduces the number of PPs and thus has the potential of power saving. This Technique used for filters implementation also for low power, fast and area efficient MAC design as an extension to this paper. The hardware implementations of the approximate multiplier including one for the unsigned and two for the signed operations can be done. It can be downloaded into FPGA for further improvements and observations.

REFERENCES

[1] S. Shafiulla Basha, Syed. Jahangir Badashah. The area efficient and spped operation of Arithmatic operations using minimal partial product' International Journal of Advances in Engineering & Technology, July 2012 ISSN: 2231-1963. liers Based on Modified Booth Algorithm'

[2]International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 1, January February 2013, pp.1513-1516,

[3] S. Jagadeesh, S. Venkata Chary, Design of Parallel Multiplier–Accumulator Based on Radix-4 Modified Booth Algorithm with SPST' International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, issue 5, September-October 2012, pp.425-431

[4] A. R. Cooper, ―Parallel architecture modified Booth multiplier,|Proc. Inst. Electr. Eng. G, vol. 135, pp. 125–128, 1988 M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[5]S. Waser and M. J. Flynn, Introduction to Arithmetic for Digital Systems Designers. New York: Holt, Rinehart and Winston, 1982.A. R. Omondi, Computer Arithmetic Systems. Englewood Cliffs, NJ:Prentice-Hall, 1994.

[6] Wang, W., Huang, X.: 'FPGA implementation of a large-number multiplier for fully homomorphic encryption'. IEEE Int. Symp. on Circuits and Systems (ISCAS), Beijing, China, May 2013, pp. 2589–2592

[7] Wang, W., Huang, X., Emmart, N., et al.:'VLSI design of a large-numbermultiplier for fully homomorphic encryption', IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., 2014, ***22****, (9), pp. 1879– 1887*

[8] Gao, S., Al-Khalili, D., Chabini, N.: 'Efficient scheme for implementing largeize signed multipliers using multigranular embedded dsp blocks in FPGAs', Int. J. Reconfigurable Comput., 2009.

[9] de Dinechin, F., Pasca, B.: 'Large multipliers with fewer DSP blocks'. Int. Conf. on Field Programmable Logic and Applications, Prague, CzechRepublic, August 2009, pp. 250–255

[10] Davis, J.P., Buell, D.A., Devarkal, S., et al.: 'High- level synthesis for largebit-width multipliers on fpgas: a case study'. Int. Conf. on Hardware/Software Codesign and System Synthesis, Jersey City, NJ, USA, September 2005, pp.213–218

[11] Baugh, C.R., Wooley, B.A.: 'A two's complement parallel array multiplicationalgorithm', IEEE Trans. Comput., 1973, ***C-22****, pp. 1045–1047*

[12] Wallace, C.S.: 'A suggestion for a fast multiplier',IEEE Trans. Electron.Comput., 1964, ***EC-13****.*

[13] Weste, N., Harris, D.: 'CMOS VLSI design: a circuits and systemsperspective' (Addison-Wesley Publishing Company, USA, 2010, 4th edn.)