VLSI Design of Low Power Booth Multiplier

Nishat Bano

Abstract- This paper proposes the design and implementation of Booth multiplier using VHDL. This compares the power consumption and delay of radix 2 and modified radix 4 Booth multipliers. Experimental results demonstrate that the modified radix 4 Booth multiplier has 22.9% power reduction than the conventional radix 2 Booth Multiplier.

Keywords- Booth multiplier, Low power, modified booth multiplier, VHDL

1. INTRODUCTION

Continuous advances of microelectronic technologies make better use of energy, encode data more effectively, transmit information more reliable, etc. Particularly, many of these technologies address low-power consumption to meet the requirements of various portable applications [5]. In these application systems, a multiplier is a fundamental arithmetic unit and widely used in circuits.

VHDL is one of the common techniques for the digital system emergent process. The technique is done by program using certain software which performs simulation and examination of the designed system. The designer only needs to describe his digital circuit design in textual form which can remove without the effort to alter the hardware. VHDL is more preferred because this technique can reduce cost and time, easy to troubleshoot, portable, a lot of platform software support the VHDL function and high references availability. All the processes will be running using Xilinx ISE 8.2i software which means the process is simulated only without any hardware implementation.

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low- power VLSI system design [6].

Fast multipliers are essential parts of digital signal processing systems. The speed of multiplier operation is of great importance in digital signal processing as well as in the general purpose processors today. The basic multiplication principle is two fold i.e., evaluation of partial products and accumulation of the shifted partial products.

2. RADIX 2 BOOTH MULTIPLIER

- *Nishat Bano is currently pursuing masters degree program in Digital Systems in Madan Mohan Malaviya Engineering College, Gorakhpur, India. E-mail: nishat_rizvi20@yahoo.co.in*

Booth algorithm provides a procedure for multiplying binary integers in signed-2's complement representation [1]. According to the multiplication procedure, strings of 0's in the multiplier require no addition but just shifting and a string of 1's in the multiplier from bit weight 2^k to weight 2^m can be treated as $2^{k+1} - 2^m$.

Booth algorithm involves recoding the multiplier first. In the recoded format, each bit in the multiplier can take any of the three values: 0, 1 and -1.Suppose we want to multiply a number by 01110 (in decimal 14). This number can be considered as the difference between 10000 (in decimal 16) and 00010 (in decimal 2). The multiplication by 01110 can be achieved by summing up the following products:

- 2^4 times the multiplicand($2^4 = 16$)
- 2's complement of 2^1 times the multiplicand ($2^1 = 2$).
-

In a standard multiplication, three additions are required due to the string of three 1's.This can be replaced by one addition and one subtraction. The above requirement is identified by recoding of the multiplier 01110 using the following rules summarized in table 1.

Q_n	Q_{n+1}	Recoded bits	Operation performed
0	0	0	Shift
0	1	+1	Add M
1	0	-1	Subtract M
1	1	0	Shift

Table 1: Radix 2 recoding rules

To generate recoded multiplier for radix-2, following steps are to be performed:

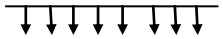
- Append the given multiplier with a zero to the LSB side
- Make group of two bits in the overlapped way

Recode the number using the above table.

Consider an example which has the 8 bit multiplicand as 11011001 and multiplier as 011100010.

Multiplicand 1 1 0 1 1 0 0 1

Multiplier 0 1 1 1 0 0 0 1 0



Recoded multiplier +1 0 0 -1 0 0 +1 -1

0 0 0 1 0 0 1 1 1

1 1 1 0 1 1 0 0 1

0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0

0 0 0 1 0 0 1 1 1

0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0

1 1 1 0 1 1 0 0 1

Product 0000001001001001

3. MODIFIED BOOTH ALGORITHM FOR RADIX 4

One of the solutions of realizing high speed multipliers is to enhance parallelism which helps to decrease the number of subsequent calculation stages. The original version of the Booth algorithm (Radix-2) had two drawbacks. They are:

(i) The number of add subtract operations and the number of shift operations becomes variable and becomes inconvenient in designing parallel multipliers.

(ii) The algorithm becomes inefficient when there are isolated 1's. These problems are overcome by using modified Radix 4.

Booth algorithm which scans strings of three bits is given below:

1) Extend the sign bit 1 position if necessary to ensure that n is even.

2) Append a 0 to the right of the LSB of the multiplier.

3) According to the value of each vector, each Partial Product will be 0, +M,-M, +2M or -2M.

The negative values of B are made by taking the 2's complement and in this paper Carry-look-ahead (CLA) fast adders are used. The multiplication of M is done by shifting M by one bit to the left. Thus, in any case, in designing n-bit parallel multiplier, only n/2 partial products are produced.

The partial products are calculated according to the following rule

$$Z_n = -2 \times B_{n+1} + B_n + B_{n-1} \quad (1)$$

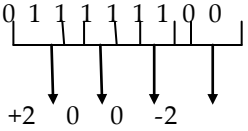
where B is the multiplier.

Table 2: Modified Radix 4 recoding rules

Consider example for radix 4:

Multiplicand 1 0 0 0 0 0 0 1

Multiplier 0 1 1 1 1 1 1 0 0



+2 0 0 -2

0000000011111110

00000000000000

000000000000

1100000010

Product 1100000101111110

4. RESULTS

We evaluate the performance of conventional and modified booth multipliers and implement them on FPGA. For Design Entry, we used ModelSim 6.5c and design with VHDL. In order to get the power report and delay report we synthesize these multipliers using Xilinx ISE 8.2i.

The comparison of synthesis report for conventional and modified Booth multi

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

pliers is given in Table 3.

Table 3

Multiplier Type	Radix 2	Radix 4
Device and family	Spartan 2	Spartan 2
No. of slices	77	72
No. of LUTs	140	129
No. of Bonded I/O	32	32
Delay(ns)	27.110	26.103
Power Dissipation(mW)	15	11

5. CONCLUSION

In this paper, the conventional and modified booth multipliers are designed using VHDL. The delay and power dissipation of modified radix 4 Booth multiplier is less as compared to the conventional one. When implemented on FPGA, it is found that the radix 4 booth multiplier consumes 22.9% less power than conventional radix 2 multiplier. Also estimated delay is less for radix 4 multiplier.

REFERENCES

[1]A. D. Booth, "A Signed Binary Multiplication Technique," *Quarterly J. Mechanical Applications in Math.*, vol. 4, part 2, pp.236-240,1951.

[2] A. R. Cooper, "Parallel Architecture Modified Booth Multiplier", *Proceedings of the Institution of Electrical Engineers,* June 1988.

[3] V. G. Oklobdzija, D. Villeger, "Analysis of Booth Encoding Efficiency in Parallel Multipliers Using Compressors for Reduction of Partial Products", *Proceedings of the 27th Conference on Signals, Systems and Computers*, pp. 781-784, 1993.

[4] Abu-Khater, Bellaouar, and M. I. Elmasry, "Circuit Techniques for CMOS Low-Power High-Performance Multipliers", *IEEE Journal of solid-state circuits, volume* 31, no. 10, October, 1996.

[5] A. P. Chandrakasan and R. W. Brodersen, Low-Power CMOS Design. *Piscataway, NJ: IEEE Press*, 1998.

[6] A. A. Fayed and M. A. Bayoumi, "A Novel Architecture for Low-Power Design of Parallel Multipliers," *Proceedings of the IEEE Computer Society Workshop on VLSI*, pp.149-154, 2001.

[7] M. 0. Lakshmanan, Alauddin Mohd Ali, "High Performance Parallel Multiplier Using Wallace-Booth Algorithm," *IEEE International Conference on Semiconductor Electronics*, pp. 433-436, 2002.

[8] Oscal T. C. Chen, et.al, "Minimization of switching activities of partial products for designing low power multipliers", *IEEE Trans. VLSI systems,* pp. 418-433, vol. 11, no. 3, June 2003.

[9] K.H. Tsoi, P.H.W. Leong, "Mullet - a parallel multiplier generator," fpl, pp.691-694, *International Conference on Field Programmable Logic and Applications,* 2005.

[10] S. K. Mangal and R. M. Badghare, "FPGA Implementation of Low Power Parallel Multiplier", *20th International Conference on VLSI Design, IEEE*, 2007.