International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 8, October - 2012

Synthesis And Simulation Of 8x8-Bit Modified Booth's Multiplier

Lakhvinder Kaur

M.Tech Student(ECE), YCOE, Punjabi University Guru kashi Campus , Talwandi Sabo – 151302 , Punjab(India)

Parminder Singh Jassal

Assistant Prof(ECE), YCOE, Punjabi University Guru kashi Campus , Talwandi Sabo – 151302 , Punjab (India)

Abstract

The booth's algorithm is the most frequently used method for multiplication. This algorithm allows the reduction in the number of partial products to be compressed in the carry adder. Thus the compression speed can be enhanced. In this paper a multiplier, using Modified Booth's multiplication algorithm is developed. After applying modified Booth's algorithm to the inputs, simple operations like two's complement or shifting plus two's complement by proceeding simple arithmetic shift are done. The number of partial products are reduced by N/2 by using this algorithm.

Key words: *Xilnx 9.2i, Model-sim XE-II 5.7g, VHDL.*

1. INTRODUCTION

A binarymultiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary numbers. The multiplier is a device which is used to perform the multiplication operation. The adders and multipliers are the basic components of designing of communication circuits. The multiplier is the basic key component of any digital signal processing system[1],[2]. Multiplication includes two basic steps, generation of partial products and their accumulation[2]. Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors and multimedia applications[3]. The type of the multiplier used for an application is based upon the requirements of the application. There are different types of multiplier available according to the requirements. The difference is in the way in which data is processed for the multiplication, the examples are serial multipliers, parallel multipliers and serial-parallel multipliers[18]. Using VHDL language a single component can be described using all three styles of modeling[22].

2. MODIFIED BOOTH MULTIPLIER

The Modified Booth multiplier is an extension of Booth's multiplier. In Modified Booth, the number of partial products reduced by N/2, that is half of total partial products as compare to simple multiplication process[4]. So, clearly if the number of partial products become reduced, the area of the multiplier also will reduced and automatically as the result of it, the speed will increased. So, this multiplier is more efficient.

3. MODIFIED BOOTH ALGORITHM

The Modified Booth algorithm is the most frequently used method to generate partial products. The partial products are reduced by N/2 by using this algorithm. So as the result of this, the multiplier can be implemented using less hardware components as compare to conventional multiplier. This algorithm can save multiplier layout area and reduces delay at the same time which are the important design advantages[1]. One of the method for high speed multiplier is to enhance the parallelism by reducing the number of calculating stages .Booths encoding reduces partial products to N/2. It converts the multiplier from radix-2 to radix-4 using redundant digit set {-2, -1, 0, 1, 2}. So in new multiplier with radix-4 there are only N/2 digits. New multiplier's 'P' digits are defined by following formula:

$Pj = Y2j + Y2j-1 -2Y2j+1$ with $Y-1 = 0$

The Modified Booth Algorithm takes following steps for multiplication:

- Modified Booth Encoder.
- Partial Product Generator.
- Sign Extension

3.1.MODIFIED BOOTH ENCODER (MBE)

Modified Booth encoding is most often used to avoid variable size partial product arrays. Before designing a MBE, the multiplier B has to be converted into a Radix-4 number by dividing them into three digits respectively according to Booth Encoder Table given afterwards. Prior to convert the multiplier, a zero is appended into the Least Significant Bit (LSB) of the multiplier. approach Instead of eight partial products being generated using conventional multiplier. Table 1. shows the truth table for a Booth encoder. The encoder takes inputs Y_{N+1}, Y_N and Y_{N-1} from the multiplier bus and produces a 1 or a 0 for each operation: single, double, negative(X, 2X and Neg).

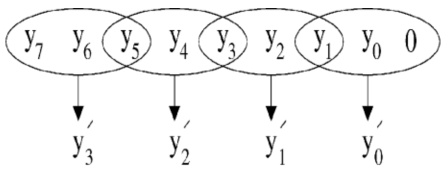

Figure 1. Grouping of bits for MBE scheme[6]

This Booth multiplier technique is to increase speed by reducing the number of partial products by half. The operand that is booth encoded is called multiplier, and the other operand is called multiplicand.

Table.1. Truth Table For Booth Encoder

Y_{N+1}	Y_N	Y_{N-1}	Operation	X	2X	Neg
0	0	0	+0 x X	0	0	0
0	0	1	+1 x X	1	0	0
0	1	0	+1 x X	1	0	0
0	1	1	+2 x X	0	1	0
1	0	0	-2 x X	0	1	1
1	0	1	-1 x X	1	0	1
1	1	0	-1 x X	1	0	1
1	1	1	-0 x X	0	0	1

In most of the cases MBE scheme is used for generating PP, because of its ability to reduce the number of PP by half[7]. The truth table shows the function of booth encoder. If a 3-bit binary input sequence is given at the input, and perform the operation as mentioned infront of it, the partial products will be generated.

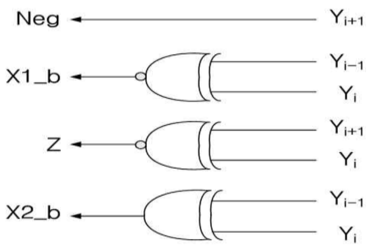

Figure 2. Booth Encoder[2]

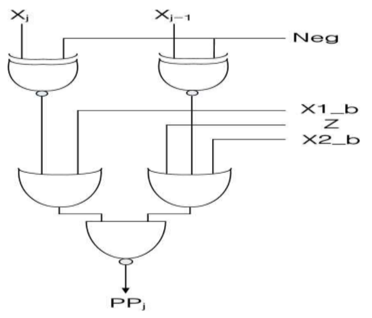

Figure 3. Booth Decoder[2]

As shown in the figure 2 and 3, the booth encoder and decoder cirrcuit is shown. As mentioned in figure 1, there are total 3 combinations for generating a partial product i.e, the obtained partial product is dependent upon the combination of three binary bits of the multiplier.

3.2. PARTIAL PRODUCT GENERATOR (PPG)

Partial product generator is the combination circuit of the product generator. Product generator is designed to produce the product by multiplying the multiplicand X by 0, 1, -1, 2 or -2. For product generator, multiply by zero means the multiplicand is multiplied by "0". Multiply by "1" means the product still remains the same as the multiplicand value. Multiply by "-1" means that the product is the two's complement form of the number. Multiply by "-2" is to shift left one bit the two's complement of the multiplicand value and multiply by "2" means just shift left the multiplicand by one place.

3.3. SIGN EXTENSION CORRECTOR

Sign Extension Corrector is designed to enhance the ability of the booth multiplier to multiply not only the unsigned number but as well as the signed number. As shown in Table 3.2 when bit 7 of the multiplicand X(X7) is zero(unsigned number) and Yn+1 is equal to one, then sign E will have one value (become signed number for resulted partial product).

Table 2. Truth Table for Sign Extension when X7 is zero.

X7	Y_{N+1}	Y_N	Y_{N-1}	E
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0

It is the same when the bit 7 of the multiplicand X(X7) is one (signed number) and Yn+1 is equal to zero, the sign E will have a new value.

Table 3. Truth Table for Sign Extension when X7 is one.

X7	Y_{N+1}	Y_N	Y_{N-1}	E
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

However when both the value of A7 and Yn+1 are equal either to zero or one, the sign E will have a value zero(unsigned number). For the case when all three bits of the multiplier value Yn+1, Yn and Yn-1 are equal to zero or one, the sign E will direct have a zero value independent to the X7 value[8].

4. RESULTS

4.1. RTL view

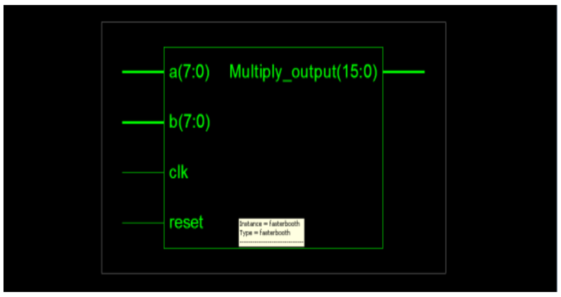

Figure 4. Top module of 8 x 8 multiplier

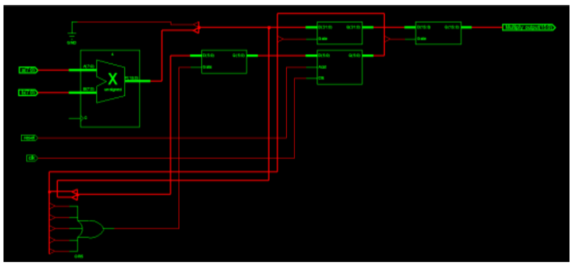

Figure 5 RTL schematic of 8 x 8 bit multiplier

As shown, ‘a’ and ‘b’ are the 8-bit inputs for the multiplier. The signals ‘clk’ and ‘reset’ are also input signals.

4.2. Simulations results

In this work, the Xilinx ISE and Model-Sim simulator are used. Here shows the outputs taken after the simulation on both of the simulators.
For the given inputs as folows:
Input a(7 to 0) : 10111011(binary)
Input b(7 to 0) : 01111011(binary)
output: 0101100111011001(binary).
The output is verified on both the simulators and that is same.

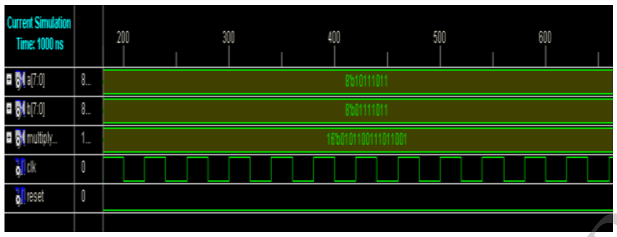

Figure 5. Simulation results of 8x8 bit multiplier for given input.

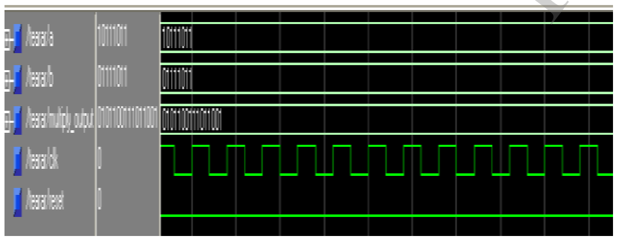

Figure 6. Simulation results of 8x8 bit multiplier for given input.

The following results are also verified.

Input a(7 to 0) : 11110001(binary)

Input b(7 to 0) : 01100111(binary)

Output : 0110011110001110(binary)

The power consumption for the design is 121 mW, with a thermal temperature of 28 ^{0}C. The table 4 shows the results for the 8-bit multiplier compared with the reference work[9].

Selected device - 3s500ecp132

Speed grade - -4

Table 4. Comparison of results

Parameter	**[9]**	**%**	**This work**	**Out Of**
Number of slices	88	1.8 %	22	4656
Number of sliced flip flops	32	0%	38	9312
Number of bonded IOB	50	54 %	33	92
Number of GCLKs	1	4%	1	4

REFERENCES

[1] C.-Y. Lee And C.W. Chiou, “Low-Complexity Bit-Parallel Multipliers For A Class Of Gf(2m) Based On Modified Booth’s Algorithm”, International Journal Of Computers And Applications, Vol. 29, No. 4, 2007.

[2] Shiann-Rong Kuang, “Modified Booth Multipliers With A Regular Partial Product Array”, IEEE Transactions On Circuits And Systems—Ii: Express Briefs, Vol. 56, No. 5, May 2009.

[3] Soojin Kim And Kyeongsoon Cho, “Design Of High-Speed Modified Booth Multipliers Operating At Ghz Ranges”, World Academy Of Science, Engineering And Technology, 2010.

[4] J. Bhasker, “A VHDL Primer”,third edition, 2008.

[5] Li-Rong Wang, Shyh-Jye Jou And Chung-Len Lee, “Awell-Structuredmodified Booth Multiplier Design, 2008 IEEE.

[6] Ki-Seon Cho, Jong-On Park, Jin-Seok Hong, Goang-Seog Choi, “54x54-Bit Radix-4 Multiplierbased On Modified Booth Algorithm”, GLSVLSI’03, April 28-29, 2003, Washington, Dc, Usa.

[7] Jung-Yup Kang, “A Simple High-Speed Multiplier Design”, IEEE Transactions On Computers, Vol. 55, No. 10, October 2006..

[8] P. Assady, “A New Multiplication Algorithm Using High-Speed Counters”, European Journal Of Scientific Research, ISSN 1450-216x Vol.26 No.3 (2009), Pp.362-368.

[9] G. Jaya Prada1, N.C. Pant, “Design And Verification Of Faster Multiplier”, International Journal Of Engineering Research And Applications (IJERA), Vol. 1, Issue 3, Pp.683-686, 2009.