Design and Verification of Faster Multiplier

G. Jaya Prada[1], N.C. Pant[2]

[1] Guru Nanak Institute of Technology, E.C.E Dept., Hyderabad, India
[2] Guru Nanak Institute of Technology, Prof, E.C.E Dept., Hyderabad, India

ABSTRACT:

The multiplier and multiplier-and-accumulator (MAC) are the essential elements of the digital signal processing. Multiplication involves two basic operations: the generation of partial products and their accumulation. Partial products can be reduced by using the Radix_4 modified Booth algorithm. The design of a binary signed-digit partial product generator, which expresses each normal binary operand in one's complement form with an extra bit denoting the sign bit of the operand. By combining multiplication with accumulation and devising a hybrid type of carry save adder (CSA), the performance was improved. Since the accumulator that has the largest delay in MAC was merged into CSA, the overall performance was improved. The CSA tree has the modified array for the sign extension in order to increase the bit density of the operands.

The parallel multipliers like radix 4 modified booth multiplier do the computations using lesser adders and lesser iterative steps. This is very important criteria because in the fabrication of chips and high performance system requires components which are as small as possible. The Multiplier and Accumulator can be adapted to various fields requiring high performance such as signal processing areas. Modelsim is used for logical verification, and further synthesizing it on Xilinx-ISE tool using target technology

Keywords: **Carry save adder (CSA), Digital signal processing (DSP), Modified booth algorithm (MBA), Multiplier and accumulator (MAC).**

I. INTRODUCTION

The digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) [1] or discrete wavelet transform (DWT) [2]. Because they are basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication and addition arithmetic's determines the execution speed and performance of the entire calculation. For high-speed multiplication, the modified radix-4 Booth's algorithm (MBA) is commonly used. This multiplier mainly consists of the three parts: Booth Encoder, to compress the partial products, and final adder. The most effective way to increase the speed of a multiplier is to reduce the number of the partial products because multiplication proceeds a series of additions for the partial products. To reduce the number of calculation steps for the partial products, MBA algorithm has been applied.

II. ARCHITECTURE OF A MULTIPLIER

A multiplier can be divided into three operational steps:

i. Radix-4 Booth algorithm in which a partial product is generated.
ii. Carry save adder and Accumulator
iii. The final addition in which the final multiplication result is produced by adding the sum and the carry.

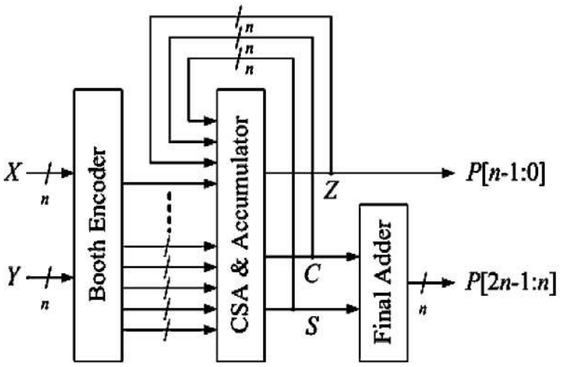


Fig1: Architecture of MAC

Generally if N-bit data of multiplicand 'X' is multiplied with N-bit multiplier 'Y' then it generates N- partial products. But if Radix-4 booth algorithm is used then number of partial products will be reduced to N/2. In addition, the signed multiplication based on 2's complement numbers is also possible.

$$X = -2^{N-1}x_{N-1} + \sum_{i=0}^{N-2} x_i 2^i, \qquad x_i \in 0,1.$$

$$X \times Y = \sum_{i=0}^{N/2-1} d_i 2^{2i} Y.$$

Where

$$d_i = -2x_{2i+1} + x_{2i} + x_{2i-1}.$$

$$P = X \times Y + Z = \sum_{i=0}^{N/2-1} d_i 2^i Y + \sum_{j=0}^{2N-1} z_i 2^i.$$

In CSA, the sign extension is used in order to increase the bit density of the operands. Half adder is used to generate sum and carry in CSA. The generated carry is stored in accumulator. In final adder both sum and carry is added to produce the 2N bits product.

III. RADIX-4 MODIFIED BOOTH ALGORITHM

The modified Booth algorithm reduces the number of partial products by half. We used the modified Booth encoding (MBE) scheme [3]. It is known as the most efficient Booth encoding [4] and decoding scheme. To multiply, multiplicand 'X' by multiplier 'Y' using the modified Booth algorithm. First group the multiplier bits 'Y' by three bits and encoding into one of {-2, -1, 0, 1, 2}. Prior to convert the multiplier, a zero is appended into the Least Significant Bit (LSB) of the multiplier. Table I shows the rules to generate the encoded signals by MBE scheme and Fig. 2 (a) shows the corresponding logic diagram. The Booth decoder generates the partial products using the encoded signals as shown in Fig. 2(b).

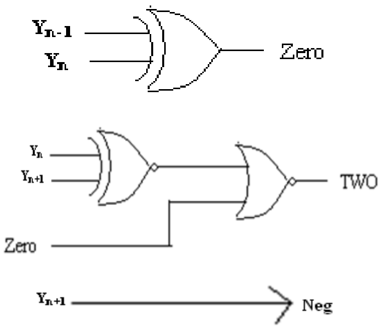


Fig.2 (a) Booth encoder

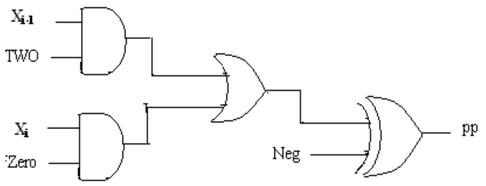


Fig.2 (b) Booth decoder

Table1: Truth table for modified booth encoder.

Y_n +1	Y_n	Y_{n-1}	Z_n	Operation	Neg	Zero	Two
0	0	0	0	0	0	0	0
0	0	1	1	1×Multiplicand	0	1	0
0	1	0	1	1×Multiplicand	0	1	0
0	1	1	2	2×Multiplicand	0	0	1
1	0	0	-2	-2×Multiplicand	1	0	1
1	0	1	-1	-1×Multiplicand	1	1	0
1	1	0	-1	-1×Multiplicand	1	1	0
1	1	1	0	0	0	0	0

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. Since the amount of hardware and the delay depends on the number of partial products to be added, this may reduce the hardware cost and improve performance.

IV. SIGN EXTENSION SCHEME

The CSA uses 1's complement and sign extension scheme [5]. 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 2 when bit 7 of the multiplicand X(X7) is zero(unsigned number) and Y_{n+1} is equal to one, then sign E will have one value (become signed number for resulted partial product). It is the same when the bit 7 of the multiplicand X (X7) is one (signed number) and Y_{n+1} is equal to zero, the sign E will have a new value.

TABLE 2 (A) Sign E 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

TABLE 2 (B) Sign E 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 X7 and Y_{n+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 Y_{n+1}, Y_n and Y_{n-1} are equal to zero or one, the sign E will direct have a zero value independent to the X7 value. The table for the Sign Extension Corrector is shown below.

Carry save adder:

The summing of the partial products in parallel using a tree of carry save adder. Half adders are used to implement the Carry save adder.

First of all the partial products should be arranged such as. The second partial product had to be shifted by two bits before adding to the first partial product. The third partial product will be shifted left by four bits where as fourth partial product will be shifted by six bits. After rearrangement partial products will be added.

V. RESULTS AND DISCUSSIONS

The multiplier code is written in verilog HDL. Multiplier contains different modules such as booth encoder, partial product generator, half adders, CSA & top module of multiplier with test bench.

To prove the correctness of our design, the verilog HDL description was simulated & tested using Model Sim Verilog Simulator and Xilinx. Xilinx Web pack is used for design entry, synthesis, place & route and floor plan design.

The input data is 8bit multiplicand and 8 bit multiplier i.e.

Multiplicand = 8'b11100111;
Multiplier = 8'b00011110;
The output generated is: 1111110100010010

Multiplicand = 8'b00101101;
Multiplier = 8'b11110001;
The output generated is: 1111110101011101

Multiplicand = 8'b00010100;
Multiplier = 8'b11011101;
The output generated is: 1111110101000100

Top Module:

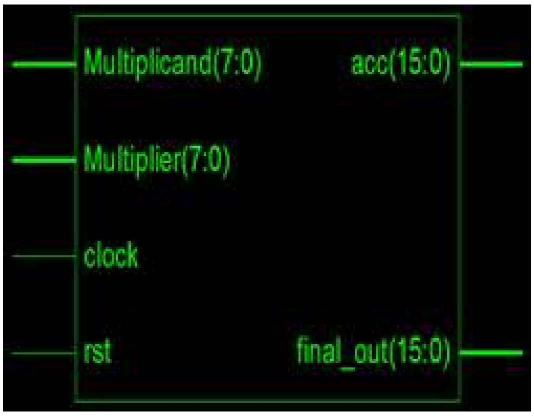


Fig 4: Top Module-Symbol of Multiplier

RTL schematic for top module:

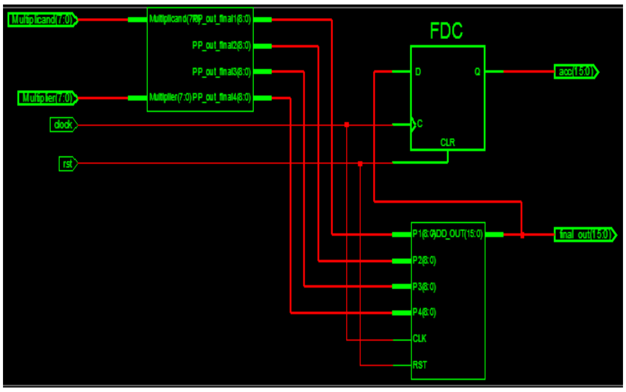


Fig 5: RTL schematic for top module

Top Module Waveform:

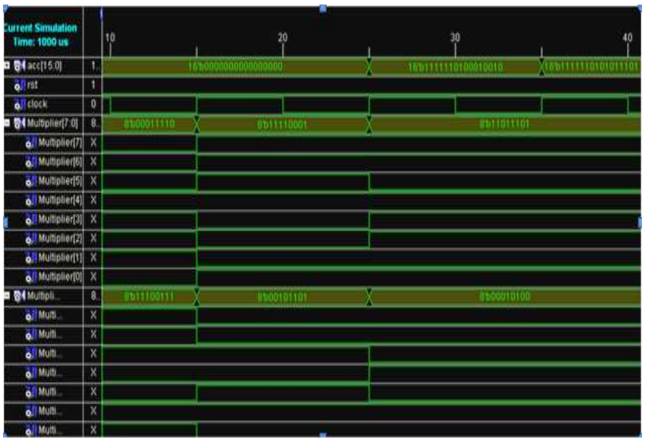


Fig.3: simulation wave form of Top Module

SYNTHESIS REPORT:

Device utilization summary:

Selected Device: 3s500ecp132-4

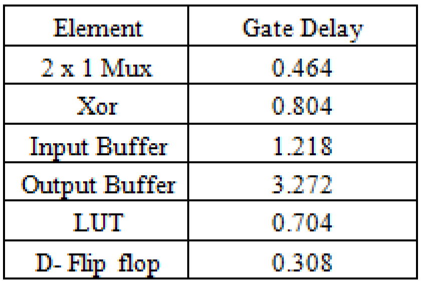
Number of Slices:	88 out of	4656	1%
Number of Slice Flip Flops:	32 out of	9312	0%
Number of 4 input LUTs:	154 out of	9312	1%
Number of IOs:	50		
Number of bonded IOBs:	50 out of	92	54%
IOB Flip Flops:	16		
Number of GCLKs:	1 out of	24	4%

Timing Summary:

Speed Grade: -4
Minimum period: 1.319ns (Maximum Frequency: 758.150MHz)
Minimum input arrival time before clock: 21.181ns
Maximum output required time after clock: 4.283ns

The Delay Summary Report:

The AVERAGE CONNECTION DELAY for this design is: 1.426
The MAXIMUM PIN DELAY IS: 4.823



Element	Gate Delay
2 x 1 Mux	0.464
Xor	0.804
Input Buffer	1.218
Output Buffer	3.272
LUT	0.704
D- Flip flop	0.308

VI. CONCLUSION

A new partial product generation technique for Booth multipliers has been proposed by eliminating the carry propagation delay encountered in generating the negative partial products in two's complement form. The modified booth algorithm is used to generate N/2 partial products. By using radix-2 algorithm the frequency is 200MHz. The proposed architecture of 8-bit multiplier using radix-4 algorithm has an operating frequency of 758.150 MHz. This can be used in the application systems requiring very high performance. The overall performance was improved.

VII. ACKNOWLEDGEMENTS

I would like to articulate my profound gratitude and indebtedness to Assoc. Prof Venkataiah, Assoc. Prof Uma rani, Asst. Prof Naga kishore, Asst. Prof Swetha for guiding and encouraging me in all aspects.

I wish to extend my sincere thanks to Asst. Prof Naga kishore for giving support.

REFERENCES

[1] Information Technology-*Coding of Moving Picture and Associated Audio, MPEG-2 Draft International Standard*, ISO/IEC 13818-1, 2, 3, 1994

[2] *JPEG 2000 Part I Final1191 Draft*, ISO/IEC JTC1/SC29 WG1.

[3] Wen-Chang Yeh and Chein-Wei Jen, "*High-speed Booth encoded parallel multiplier design," IEEE Trans. on Computers, vol. 49, issue 7,* pp. 692-701, *July 2000.*

[4] F. Elguibaly, "*A fast parallel multiplier–Accumulator using the modified Booth algorithm*," *IEEE Trans. Circuits Syst.*, *vol. 27, no. 9, pp.* 902–908, Sep. 2000.

[5] M. D. Ercegovac and T. Lang, *Digital Arithmetic* (Morgan Kaufmann Publishers, Los Altos, CA 94022, USA, 2003)

[6] Yung-chin Liang, Ching-ji Huang and Wei-bin Yang, "*A 320-MHz 8bit x 8bit pipelined multiplier in ultra-low supply voltage*," Proc. of IEEE A-SSCC, pp. 73-76, *Nov. 2008.*

[7] O. L. McSorley, "*High-speed arithmetic in binary computers*," Proc. of Institute of Radio Engineers (IRE), *vol. 49, no. 1,* pp. 67-91, 1961