Design and Analysis of Booth Multiplier with Optimised Power Delay Product

Ch V S Chaitanya, PSathish Kumar
Department of Electronics and Communication Engineering,
Amrita School of Engineering Bangalore, Amrita VishwaVidyapeetham, India
Email: chaitu.chokkara93@gmail.com, p_sathishkumar@blr.amrita.edu

***Abstract*—In most of the VLSI systems, multiplier being the vital part consumes nearly 15-20% of total IC power and is quiet slow in overall operation of the system. Thus it is essential to have an efficient design for the multipliers to improve the overall performance of the system. Booth multiplier reduces the number of partial products, taking into account two bits of the multiplier at a time, resulting in speed advantage over other multiplier architectures. With this advantage, Booth Multiplier is widely used in multiplication process for various digital and DSP circuits. The objective of this paper is to implement an optimized Booth Multiplier (8*8) with improved Power consumption and Delay Product (PDP). The sign extension is implemented using a single inverter and the addition operation is implemented by using custom designed Carry Skip Adders with 10T Full Adder. The design implementation and the simulations are done in Cadence Virtuoso V13.0 under 45nm technology.**

***Index Terms*—**Booths Algorithm, Booths Encoder, Booths Decoder, 10T full adder, Carry Skip Adder

I. INTRODUCTION

With the advent of deep submicron technologies, the operating speed among all the three design parameters, have a tradeoff with both Area and Power consumption. Thus the optimization of operating speed is essential for circuit's efficiency. The operating speed is a function of two factors.

1. Circuit technology
2. The used algorithm.

Multipliers are the integral components in almost all the complex digital circuits. Thus, the performance of the electronic device is predicted generally by the operating performance of the Multiplier in the system, which is the most area consuming. Hence, optimizing the speed and area of the multiplier is an important task for better performance of the circuit[2]

The multiplication process is basically addition of a number with multiplier no of times to get the multiplication output. This requires mainly 2 steps:

1. Partial product generation
2. Addition of generated Partial Products

This process consumes more area and exhibits more delay. There is a similar process where the generation of Partial Products is carried by shifting the multiplicand with comparison to a significant booth bits. This is the Booths Algorithm introduced by "Andrew Donald Booth", [1] [7]. The algorithmic flow chart of Booth Multiplication is shown below in Fig 1.

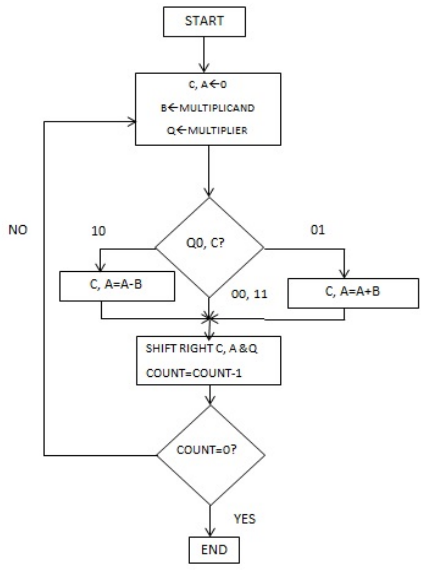

Fig 1 Booths algorithm for multiplication

BOOTHS MULTIPLIER

Booth's algorithmworks on the principle of encoding the multiplier, shifting the multiplicand, rearrange at the appropriate bits and form the partial products. This scheme of operation is different and effective in reduction of delay in the system, when compared to other multiplication techniques. Fig 2shows the block diagram Architecture for implementing booths algorithm in multiplication [1][7].

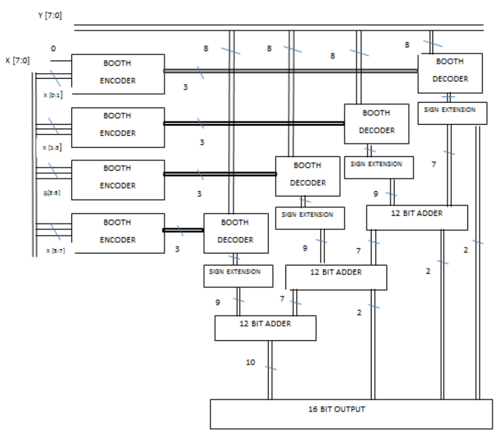

Fig 2 Booths Architecture

The design of booths multiplier consists of three individual modules:

[i] BOOTHS ENCODER:

Booth's algorithm involves repeatedly addingone of two predetermined values to a number, and then performing a rightward arithmeticshift on the number. This operation is done by the encoder.The input (multiplier) is given through the encoder. The input is divided into groupings of 3 bits and subjected to radix-2 encoding scheme to generate single (x), doubled (2x) and a negative (-x) selection bits for the multiplicand to operate. Table1 shows the truth table for Booth encoder and Fig 3 shows the design implementing the encoder truth table.

Table 1 Truth table for booths encoding

X2i +1	X 2i	X2 i-1	operat ion	Sing le	Dou ble	Negat ive
0	0	0	0	0	0	0
0	0	1	Y	1	0	0
0	1	0	Y	1	0	0
0	1	1	2Y	0	1	0
1	0	0	-2Y	0	1	1
1	0	1	Y	1	0	1
1	1	0	-Y	1	0	1
1	1	1	0	0	0	1

Encoding is the method used to reduce partial products. For an n-bit multiplication [2][8] booths algorithm examines n+1 bits of multiplier and encodes n bits. Booth encoder 4 bit is designed using 4 single bit encoder circuits as shown in Fig 4.

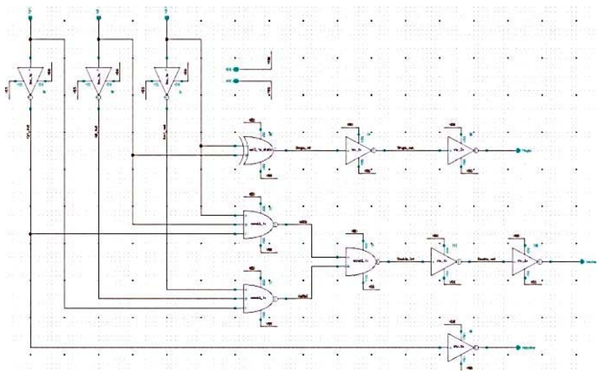

Fig 3 Single bit booth encoder

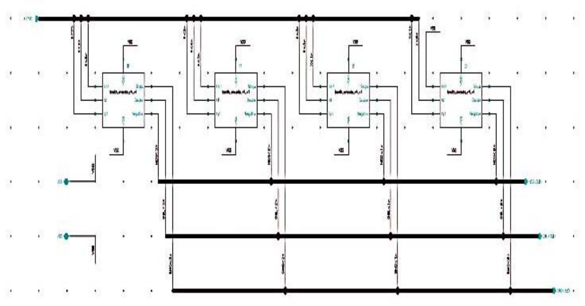

Fig 4 4bit Booth Encoder.

[ii] BOOTHS DECODER:

Table 2 shows a truth table of the decoder block for generating a single bit of a partial product with inputsfrom the encoder output bits and two of the multiplicand bits. As this truth table doesn't give partial product for negated bits, it is executed by using Xor with Partial product for all negative asserted bits.The truth table for decoder shown in Table 2is based on the Single and Double inputs. Fig5shows the gate level/logic level implementation of the single bit Booth decoder.

Table 2 Truth table for booths decoder

Single	Multiplicand (Y)	Double	2Y	Partial product
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	1	0	0	1
1	1	0	1	1

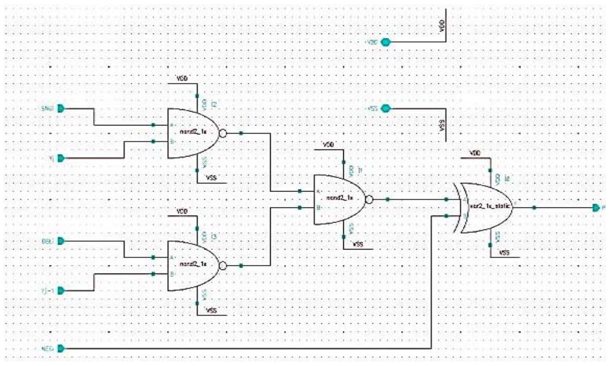

Fig 5 Single bit Booth Decoder

The complete partial product is implemented by 9bit booth decoder shown in Fig 6. This design comprises of 9 single bit booth decoders, which are linked to a half adder(HA) each, that sums the partial product outcome with previous HA carry. The first HA is asserted with Negative selection bit from encoder, this is to implement 2's complement form when the Negative bit is asserted.When Double selection bit is asserted, the shifted multiplicand (2Y) bit is grounded, so that Zero is shifted in.The ninth bit is produced by connecting the Yj-1 of 9th decoder to the MSB of multiplicand i.e., Y<7>. The use of ninth bit is:

- If shift occurs, the PP<8> is replaced by 9th bit.
- If shift doesn't occur it acts like a sign extension bit, simply no change in Y<7>

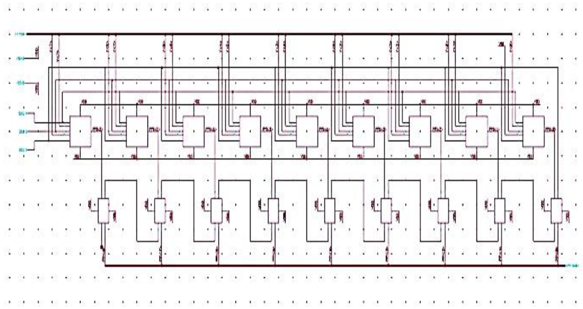

Fig 6 8bit Booths Decoder

[iii] Partial Product Addition:

The partial products thus generated are shifted and rearranged bits according to the algorithm, which are needed to be added for the complete multiplication output.In order to sum the 9-bit partial products from the decoders, a 12-bit CSkA was implemented from the 4-bit CSkA blocks. This circuit cascades four bit CSkA's in the same way that the 2 bit CSkA's cascade full adders. The implementation of carry skip logic for the 12-bit adder provides further improvement inspeed of the multiplier.

SIGN EXTENSION OF PARTIAL PRODUCTS:

For proper addition of the partial products in signed multiplication the partial products are to sign extended, without any disturbance to the functionality. This is to preserve the sign digit in the shifting oreration and rearranging the partial products for addition [2]. This process can be put in implementation by these steps[2]

- For each output of decoders, invert the MSB.
- For the first Parital product, Add 1 to the MSB.
- For all remaining Partial products, Add 1 infront of MSB.

This technique can be implemented by placing an inverter between the 9th bit of partial product and 10th adder input. The inverted outputs are tied to reduce the circuitary.10th input of each adder is connected to VDD, to add 1 infront of each Partial Product MSB. Addition of '1' infront of MSB of first Partial product generates a carry bit, which is an unwantedstate for previous cycles carry output. PP08 is the MSB of first Partial product, there are only two possibilities for the addition:

1. If PP08=1; No carry is produced from first partial product, i.e., B8B7B6=100.
2. If PP08=0; Carry is produced from first partial product, i.e., B8B7B6=011

The circuit can be simplified by tying the similar bits(B6=B7=~PP08 & B8=PP08). The design of sign extension implementation is shown in Fig 7.

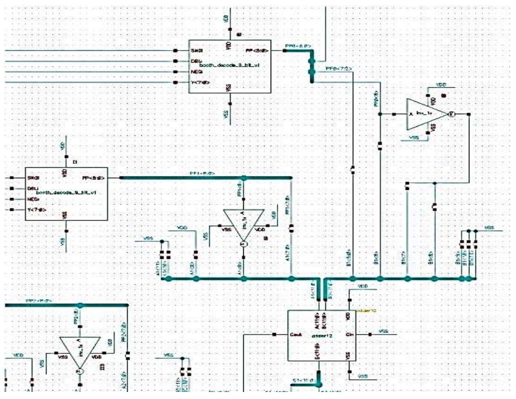

Fig 7 Sign Extension technique using inverter

II. CARRY SKIP ADDER

10T FULL ADDER:

The full adder used in this design, is implementedby the 4T Xnor logic[5] at inputs and pass transistors to propagate sum and carry. Though the output logic level is degraded by the pass transistors, the operation of 4T Xnor logic reduces the voltage degradation [9]. This is a well optimized technique with improved speed and less gate count. The implemented Full adder is shown below in Fig 8.

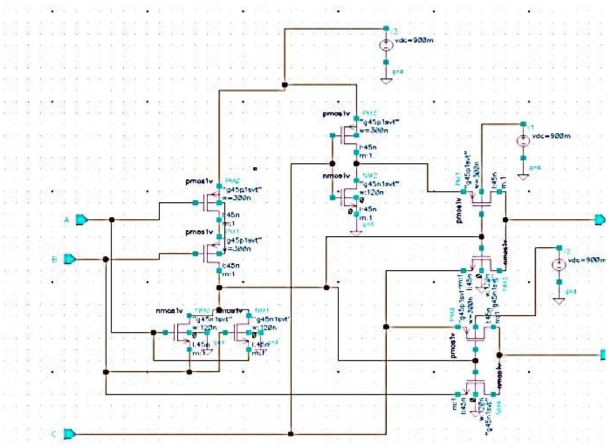

Fig 8 10T Full Adder

OPERATION AND DESIGN OF CARRY SKIP ADDER:

Carry Skip adder uses a Skip or Bypass Logic to reduce the delay in carry propagation by skipping the carry propagating path depending upon the carry generation [3].In the worst case delay of carry propagation i.e., when both the implicants are high, the input carry is propagated throughout the n-bit adder and appear at Cout. The worst case delay is given by

D (Cout)= n*D (Cin).

Propagation conditions are determined by the use of Xor gates with inputs as the operands itself. For this condition to be true, the carry Cin decides the propagating carry Cout for each addition. The n bit Carry Skip logic needs n bit AND gate and a MUX to propagate the carry.Each Xor output is connected to an n input AND gate and the output is used to select the propagating carry through the MUX[6]

The carry bit which is not needed to propagate is now skipped through an n-bit bypass where the inputs are at high state. This reduces the carry latency in individual adder blocks.Here the width of AND gate is taken equal to that of the Adder. This is to regularize the selection of carry for entire length of the adder [3]. Fig 9 shows the schematic of 2bit Carry Skip Adder.

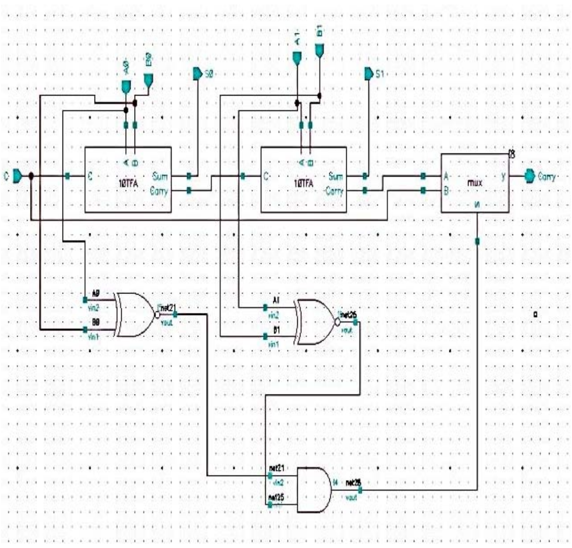

Fig 9 2bit Carry Skip Adder

Fig 10gives the design implementation of 12bit Carry Skip Adder.The designed 2bit CSkA has shown the exact expected output with a reduced delay of 3.02 ns. The implementation of carry skip logic with Xnor and And gates, consumes more transistors on chip, thereby increasing the Design Area of Adder.

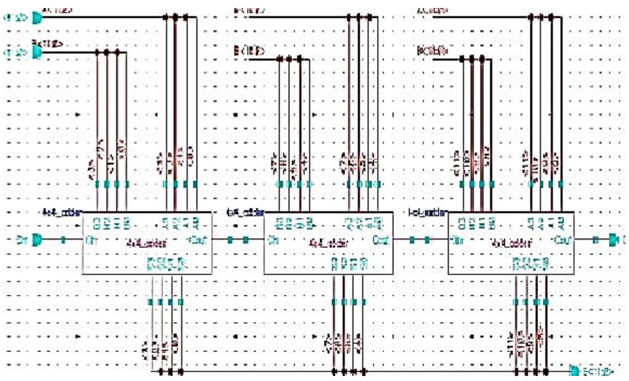

Fig 10 12bit Carry Skip Adder

The final schematic of the modified Booth multiplier 8*8 bit is shown in Fig 11. The designed Booths multiplierproduces a 16-bit output equal to the product of the two 8bit inputs

III. SIMULATION AND RESULTS

The simulation results are observed for signed and unsigned test vectors with different delay at the output. The results are shown in Fig12 (a),(b)&(c). The delay and power consumption details are shown in Table3.The results are compared with booths multiplier designed using conventional full adder and 12bit CLA.

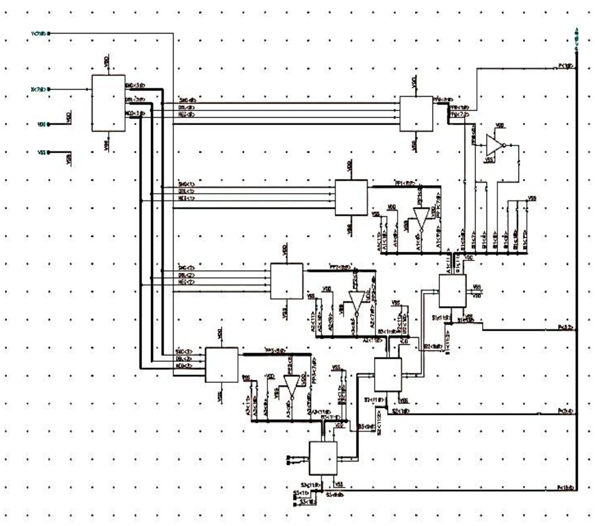
Fig 11Booth Multiplier Schematic

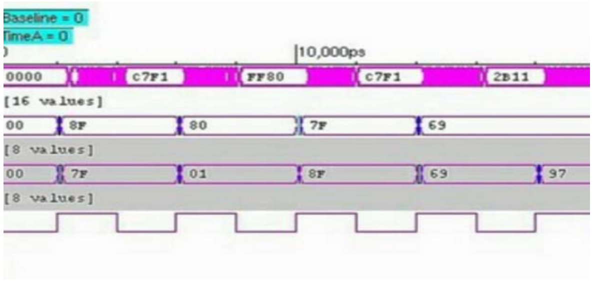

Fig12(a) Simulation Results

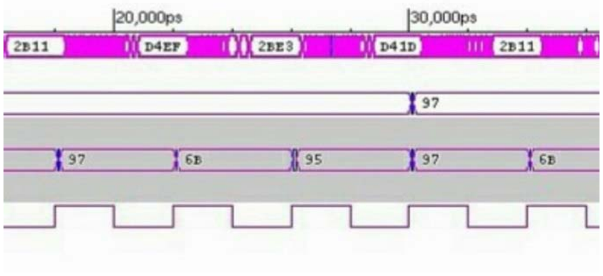

Fig12 (b) Simulation Results

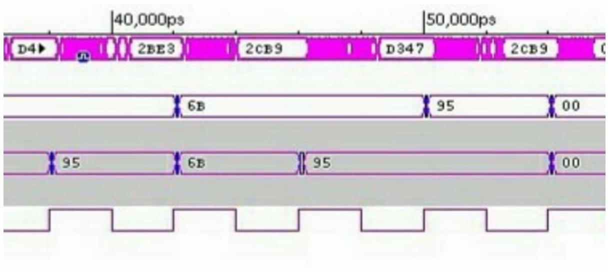

Fig12 (c)Simulation Results

Table3 Delay and power consumption comparison.

Parameter	CSkA 12bit	CLA 12bit	Booth Multiplier	Optimised Booth Multiplier	%change in parameter in Optimised design
Delay (ns)	6.98	8	72.1	53.4	-25.9
Power consumed (nW)	103.75	78.4	168.95	212.38	+25.16
PDP	724.175	627.2	12188.3	11341.09	-6.89

IV. CONCLUSION

The analysis of the design of Optimized Booth Multiplier 8*8 bit shows the overall improvement in the PDP (power delay product) by 6.89% than the existing Booth Multiplier designed with conventional full adder and CLA. There is an increase in power consumption of 25.16% than the existing design; this is presumed to be because of multiplexers used in the Carry Skip Logic in the 12bit adder used. The delay is optimized to 25.9%. The increase in no of transistors on chip due to Carry Skip Logic is feasible with the delay reduction.

Thus the overall design is considered to be satisfactoryin optimization of PDP. The power consumption and area optimization of the designed circuit can be considered for further study.

V. REFERENCES

[1] Ramya Muralidharan, Chip-Hong Chang, "Radix-4 and Radix-8 Booth Encoded Multi-Modulus Multipliers", IEEE Trans on Circuits and System, no. 11, Nov 2013.

[2] Wang Jiun-Ping, Shiann-RongKuang, Shish-Chang Liang, "High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications", IEEE Transaction on VLSI system, vol. 19, no. 1, Jan 2011.

[3] M. Alioto, G. Palumbo, "A Simple Strategy for Optimized Design of One-Level Carry-Skip Adders", IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol. 50, no. 1, January 2003.

[4] Performance Analysis of a Low-Power High-Speed Hybrid 1-bit Full Adder Circuit: Partha Bhattacharyya, Bijoy Kundu, Sovan Ghosh, Vinay Kumar, and Anup

Dandapat, IEEE transactions on very large scale integration systems, vol. 23, no. 10, October 2015.

[5] Performance and Analysis of 10T Full Adder Using MTCMOS Technique: Vinod Agarwal, Ravi Shrivastava, Shyam Akashe ITM University Gwalior 2015 International Conference on Communication Networks.

[6] C. N. Nagendra, M. J. Irwin, R. M. Owens, "Area-Time-Power tradeoffs in parallel adders", IEEE Trans. Circuits and Systems XII: Analog and Digital Signal Processing, vol. 43, no. 10, Oct. 1996.

[7] S. Sri Sakthi, N. Kayalvizhi, "Power aware and high speed reconfigurable modified booth multiplier", IEEE Recent Advances in Intelligent Computational Systems, 2011

[8] P. R. Gokul, E. Prabhu, H. Mangalam, "Performance comparison of multipliers based on Square and Multiply and montgomery algorithms", International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE) 2014.

[9] Rajesh KannanMegalingam, GauthamPopuri, ParthasarathyRavisankar, "Low Power Consumption Coarse Grained Reconfigurable Adder", Second International Conference on Computer and Electrical Engineering, 2009.