An Efficient Implementation of Wallace Tree Multiplier

Anagh P. Deshpande
Mtech:Very large Scale Integration(VLSI) design
School of Electronics Engineering(SENSE)
VIT University, Chennai
anagh.deshpande22@gmail.com

Ameya A. Chandras
Mtech:Very large Scale Integration(VLSI) design
School of Electronics Engineering(SENSE)
VIT University, Chennai
ameyaa.chandras@gmail.com

Dr. Vigneswaran T.
Professor
School of Electronics Engineering(SENSE)
VIT University, Chennai
vigneswaran.t@vit.ac.in

Abstract— In Very large Scale Integration (VLSI) technology, power consumption and speed are the two important constraints for determining the efficiency of the architecture. This paper aspires at declining this parameters of the Wallace tree multiplier with the efficient use of modified Booth encoding and compressors.The proposed architecture is employed in Verilog HDL and it is simulated in Cadence NC Sim and synthesized using Encounter RTL Compiler in 180nm Taiwan Semiconductor Manufacturing Company(TSMC) slow library .The proposed architecture is found to be 42.2% faster than the conventional Wallace tree architecture and the power consumption lowered by 45% as compared to the conventional Wallace Tree.

Keywords*—Wallace tree; Booth Encoder; Compressor;*

*****

I. Introduction

Multipliers are extremely vital in any basic algorithmic computation. Many researchers have done stupendous work on the design of increasingly efficient multipliers which plays a significant role in applications like digital signal processing system where effective utilization of processing time with minimum latency is most sought out. They aspire to provide this with lesser power consumption. This makes them easier to implement in various composite and wieldy circuits.

This section depicts the definition of the terms which are introduced in this paper. Section 2 goes over the traditional approaches of Wallace tree architecture. Section 3 comprises of modified Wallace architecture for multiplier reduction. Section 4 compares the results of modified Wallace tree architecture with typical one.Section 5 summarizes the conclusions.

Multipliers comprises of ternary stages viz formation of partial products,accumulation and downsizing of partial products and final addition of the last stage.The first stage is simply the AND operation of each bit of the multiplier and multiplicand. Booth encoding can be used in this stage by which we can decrement the number of partial products. The second stage is decrement in the number of partial products which is utmost vital as it is more intricate and that decides speed of whole multiplier. This stage normally conduces the most of the latency, power and area of the multiplier. The final stage contains various adders to generate the final output by summing up two rows of partial products. The typical Wallace tree multiplier employ carry save adders to cut down N-row bit product matrix to a commensurate two row matrix [1]. The obtained matrix, then is added up by carry look ahead adder to yield the output.In the regard of adders in the intermediate stage, the conventional Wallace tree uses full adders and half adder[2].This paper is all about reduction of delay in the computation time of multiplication with lesser power consumption. For the above mentioned purpose, compressors are utilized in the architecture of the multiplier.

II. Background Study

A . Conventional Wallace Tree

Conventional Wallace tree is an excellent approach to reduce partial product reduction. It was first proposed by C.S Wallace in 1964[2]. It was then used by many research scholars for above purpose. Instead of directly adding partial products, it suggest that once partial product array is formed, we can arrange adjacent rows intonon-imbrication groups of ternary. Each group is then reduced by either applying full adder or half adders depending on input bits and then moving any single bits to the next stage. This kind of process is carried out for each stage until only two rows are remaining. Then finally those row are added by Ripple arry adder. This process is demonstrated by 8 bit by 8 bit typical Wallace tree depicted by Fig. 1 .

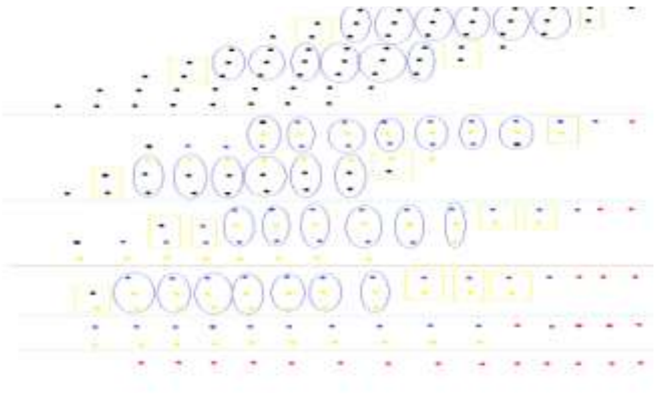

Figure 1. Conventional Wallace Tree

III. Proposed Architecture

The architecture of the proposed Wallace tree multiplier using Booth encoder and 3:2 compressorcompromises of four blocks of operation.In the following subsections,every component is explained.

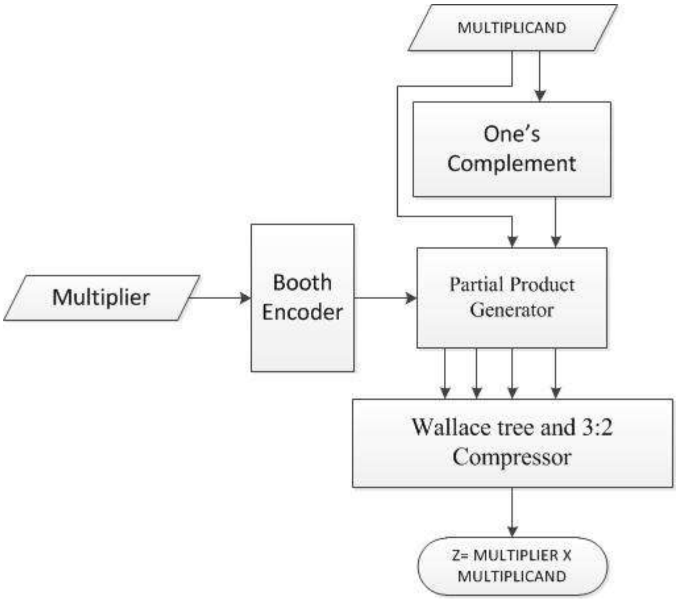
MULTIPLICAND
One's Complement
Multiplier
Booth Encoder
Partial Product Generator
Wallace tree and 3:2 Compressor
Z= MULTIPLIER X MULTIPLICAND

Figure 2. Proposed Architechure

A. One's complement generator

For the multiplication of signed numbers ,it is utmost important for complementing the multiplicand i.e. one's complement generator. This can be achieved by generating inverted bits(changing 1 to 0 and vice-a-versa) of the multiplicand. One's complement is nothing but the negative of the original multiplicand. It is essential for the partial product generator to easily distinguish between multiplication of both signed as well as unsigned multiplication.

B. Booth Encoder:

Traditionally, multiplication includes calculation of partial products by ANDing operation and then subsequently adding all those partial products. When we multiply a N bit number by another N bit number, the total number of partial products generated equals to N^2.In the architecture, modified booth encoder is used for reduction of partial produts and to speed up the computation of the partial product array. Thus, the number of partial products can be reduced from N^2 to (N/2 -1).Booth encoder enables the multiplication of positive as well as negative numbers.[3]Let A and X be two N bit numbers where A is the multiplicand and X is the multiplier.

They can be represented by equations (1) and (2) respectively.

$$A=-a_{N-1}2^{N-1}+\sum_{i=0}^{N-2} a_i \cdot 2^i \quad (1)$$

$$X=-x_{N-1}2^{N-1}+\sum_{i=0}^{N-2} x_i \cdot 2^i \quad (2)$$

where $a_i$is the i-th bit of multiplicand and $x_i$is the i-th bit of the multiplier[4].

In the proposed architecture, we are using radix -4 modified Booth encoder to encode the multiplier bits. The radix-4 modified Booth encoding includes scanning the multiplier with a window size of 3bits or simply called as scanning of triplets. The grouping of the 3bits can be done as shown in the Figure 2.

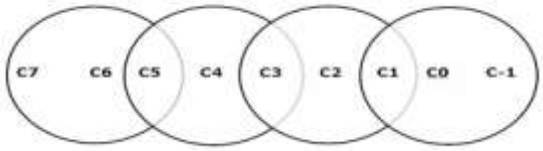

Figure 3. Grouping of 3 bits of multiplier for Booth encoding

Let $$X=\sum_{i=0}^{MD/2-1} M_i \cdot 2^{2i} \quad (3)$$

$$X=\sum_{i=0}^{MD/2-1} (-2x_{2i+1}+x_{2i}+x_{2i-1})\cdot 2^{2i} \quad (4)$$

In the above equation (3), the possible values of Mi can be {-2,-1,0,+1,+2}.

The product can be represented as follows:

$$Z=A\times X=\sum_{i=0}^{2MD-1} P_i \cdot 2^i \quad (5)$$

where, Pi is the product bit at i-th iteration which can be represented as

$$Z=\sum_{i=0}^{(MD-2)/2} (-2x_{2i+1}+x_{2i}+x_{2i-1})\cdot 2^{2i} A \quad (6)$$

$$Z=\sum_{i=0}^{(MD-2)/2} S_i \quad (7)$$

where, Si is grouping of 3bits of the multiplier starting from LSB to MSB by overlapping the significant bit of the previous grouping.

Considering N=8, the possible values of Mi are {-2,-1,0,+1,+2}.

The modified Booth encoding values are as shown in the following table.

TABLE I. BOOTH ENCODER VALUES

$D2_{(x2i+1)}$	$D1_{(x2i)}$	$D0_{(x2i-1)}$	Operation	Neg	Two	One	Zero	Cor
0	0	0	+0	0	0	0	1	0
0	0	1	+X	0	0	1	0	0
0	1	0	+X	0	0	1	0	0
0	1	1	+2X	0	1	0	0	0
1	0	0	+2X	1	1	0	0	1
1	0	1	-X	1	0	1	0	1
1	1	0	-X	1	0	1	0	1
1	1	1	-0	1	0	0	1	0

(-X) denotes the 2's complement of the number which can be obtained by inverting the bits of X and adding 1 to the LSB. (-2X) can be obtained by doing left shift operation on (-X) by one bit and putting 0 at the LSB position.

(+2X) can be obtained by doing left shift operation on (X) by one bit and putting 0 at the LSB position.

The equivalent circuit of the Booth encoder can be designed from the values obtained in the Table 1 as shown in figure 3

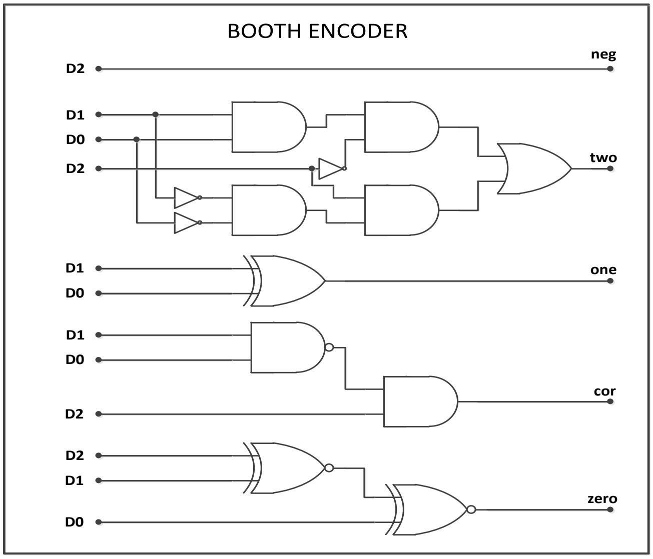

Figure 4. Booth encoder circuit

The D2,D1,D0 are the 3bits of the multiplier which are fed as the input to the Booth encoder. The neg bit will notify whether a negative number has been multiplied to the multiplicand or not. Zero bit tells whether it is to be considered as partial product or not. One bit indicates the direct use of multiplicand and two bit indicates whether left shift operation is needed or not and the cor bit is the error correction bit which is to be summed with the last partial product.

C. Partial Product Generator

Partial product generator is designed by the use of multiplexers as shown in figure(5).The multiplicand and the complement of the multiplicand are the two inputs which are fed to the multiplexer and the neg acts as a select signal for it. The next stage multiplexer has combination of two, one and zero bits as the select signal. Any one of them will be high at a time. The previous state of the input will be given as output if two is high, the output of the multiplexer in the first stage will be given as the input to the multiplexer in the second stage and the output will be 0 if the zero bit is high. After one iteration, one partial product bit will be produced. The whole logic can be repeated for the generation of all partial product bits.

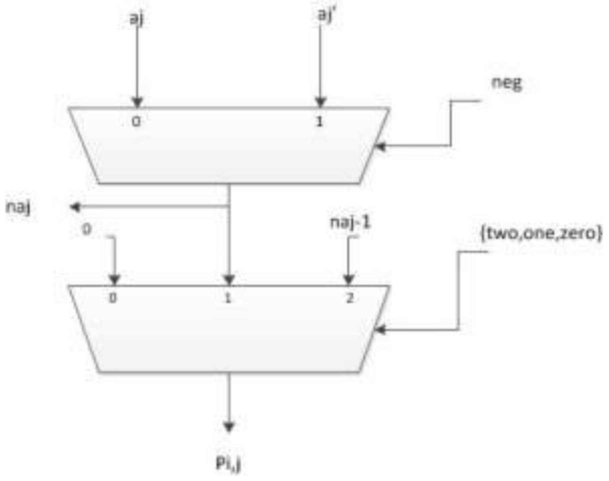

Figure 5. Partial Product Generator

D. Wallace tree using 3:2 Compressor

The conventional Wallace tree uses half adders and full adders as carry save adders. But this leads to increased delay in the computation of partial products. In order to reduce this latency, the full adders are replaced with 3:2 compressors 3:2 Compressors has XOR-XNOR and MUX as building blocks. So delay of this device is ΔXOR+ ΔMUX [5] .

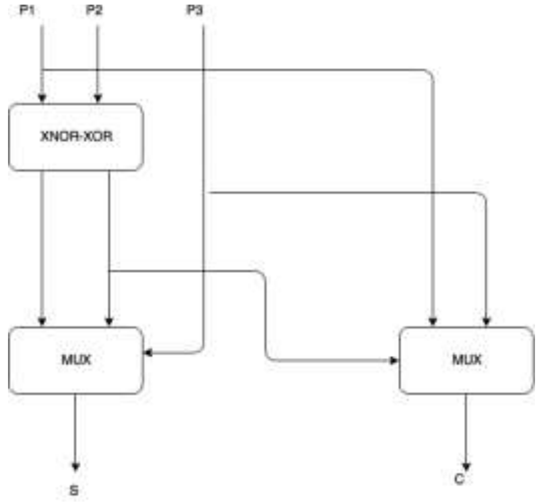

Figure 6. 3:2 Compressor

P1,P2,P3 are the three inputs of the 3:2 compressors whereas SUM and CARRY are the ouputs of the 3: 2 compressors[6]. The structure can be represented using,

S = (P1^P2)*(P3)’ + (P1^P2)’*(P3) (6)

C=(P1^P2)*(P3)+(P1^P2)’*(P1)(7)

IV. SIMULATION AND RESULTS

RTL schematic for Wallace tree multiplier using modified Booth encoding and compressor was generated successfully by the RTL compiler with the use of constraints file and output is verified by Cadence NC Sim simulation[7].

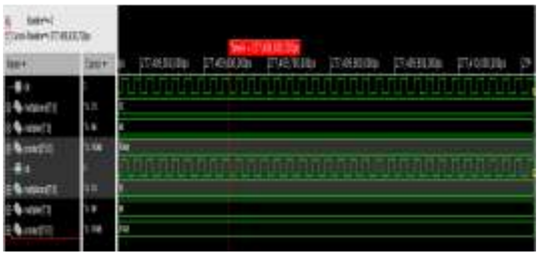

Figure 7. Simulated output waveform of conventional Wallace tree multiplier

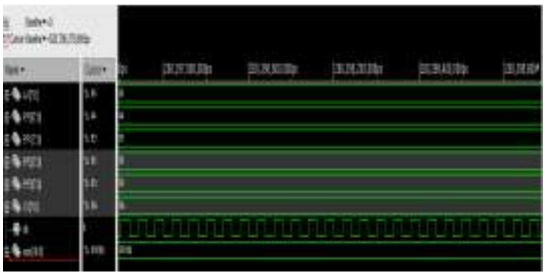

Figure 8. Simulated output waveform of Booth encoded Wallace tree without compressor

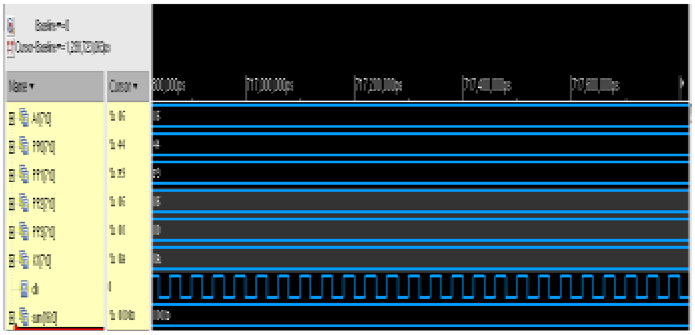

Figure 9.Simulated output waveform of the proposed Wallace tree multiplier with Booth encoding and compressor

The following tabulated parameters in the Table II below were formulated using the Cadence Encounter® RTL Compiler's power,timing and area reports respectively.

TABLE II. COMPARATIVE TABLE

	Power(nw)	Delay(ns)	Cells	Area
Conventional Wallace Tree multiplier	402062.22	9.851	284	7152
Wallace tree multiplier with Booth encoding	222072.70	6.103	266	10372
Wallace Tree multiplier with Booth encoding and Compressor	220859.02	6.924	322	10605

From the above table, we can infer that , in terms of area conventional Wallace tree multiplier performs better. In the power consumption, Wallace tree multiplier with Booth encoding and compressor achieve less value. But Wallace tree with Booth encoding works faster than the rest of the architecture.

V .CONCLUSION

The proposed architecture is better than conventional Wallace tree by 45% in terms of power consumption. The speed of Wallace tree multiplier with Booth encoding and compressor is better by 42.2% than conventional Wallace tree . The delay is reduced by the use of Booth encoding and compressor. To further reduce power consumption and to improve the speed of architecture ,4:2 compressor can be used[8] [9].

REFERENCES

[1] Ron walters,Earl Swartzlander,"A reduced complexity Wallace multiplier reduction", IEEE Trans.,Vol 59,no 8, 2010.

[2] C.Wallace,'The suggestion of fast multitplier',IEEE Trans.Electron.Comp.vol.EC/13,pt.14/17,Feb 1964

[3] J. fadavi-Ardekani,MxN Booth encoded Multiplier Generator using optimized wallace trees, IEEE Trans.vol.1,no2, june1993.

[4] J.Pi ng Wang,Shiann-Rong Kuang, Shish- chang Liang."High accuracy fixed width modified Booth multipliers for lossy application.

[5] Shrivari Veermanchaneni, kirthi M,krishna Lingamneni Avinash Sreekanth Reddy Puppala M.B. Srininvas."Novel Architecture for High speed and low power 3-2,4-2 and 5-2 Compressors,"20th International conference on Very large scale integration design(VLSI),jan 2007.

[6] J.Rao,Sanjay Dubey,"A high Speed Wallace tree multiplier using Booth algorithm for Fast Arithmetic circuits", IOSR Journal of Electronics and Comm.Engg.(IOSRJECE) ISSN: 2278-2834,VOL 3, No. 1(sep-oct 2012),pp 07-11.

[7] Cadence TDA Tools(RTL Compiler, NC Sim)

[8] C. Vinoth1,V.S. Kanchana Bhaaskaran2,B.Brindha,S. Sakthikumaran,V.Kavinilavu,B.Bhaskar, M Kangasasabapathy and B. Sharath ,"A Novel low power and high speed Wallace tree multiplier for RISC processor", @2011 IEEE

[9] Prasad k.and Parhi K.k.,"low power 4-2 and 5-2 compressors",@Proc of 35th Asilmour Confe.vol.1 ,2001,pp. 129-133.