Analysis of Multipliers Based on Various Performance Measures

Savita Nair
PG student, Electronics, Pillai's Institute of Information Technology, New Panvel, India

Ajit Saraf
Asst. Professor, Electronics, Pillai's Institute of Information Technology, New Panvel, India

Swati Nair
PG student, Electronics, Pillai's Institute of Information Technology, New Panvel, India

Abstract- **The multiplier is the essential element in various digital signals processing such as filtering and convolution. Multiplication operation involves the generation of partial products and their accumulation. The speed of multiplication can be increased by reducing the number of partial products or by accelerating the accumulation of partial products. This study paper summarizes the comparative study of four multipliers namely, Array multiplier, Modified booth multiplier, Wallace tree multiplier and modified Booth-Wallace tree multiplier based of various performance measures.**

Keywords: Array Multiplier, Modified Booth Multiplier, Wallace tree Multiplier, Modified Booth-Wallace tree Multiplier

1. INTRODUCTION

A multiplier is one of the main blocks in most of the digital signal processing (DSP) systems. The various DSP applications where a multiplier plays an important role include digital filtering, digital communications and spectral analysis. Multipliers are complex circuits which operate at a high system clock rate, therefore, it is essential to reduce the delay of multiplier in order to satisfy the overall design. In addition, multiplier consumes large area and dissipates more power. Hence designing of multipliers that offer any of the following design targets – high speed, low power consumption, less area or even a combination of them is of great interest.

The multiplier architecture can be classified into three stages namely, partial product generation stage, partial product reduction stage and the final addition stage. The "add and shift" algorithm is the common multiplication method employed. The speed of multiplication can be increased by either reducing the number of partial products or by increasing the speed of accumulation of partial products. Multibit compressors can be used for reducing the number of partial product addition stages.

Multipliers can be classified into two categories namely, serial and parallel multipliers. In serial multiplier, each bit of multiplier is used for evaluation of partial products whereas in parallel multipliers, partial product from each multiplier bit is computed in parallel. The main parameter that determines the performance of parallel multipliers is the number of partial products that needs to be added. Speed is compromised in case of serial-parallel multipliers to achieve better performance in terms of area and power consumption. Depending on the nature of application parallel or serial multiplier is selected. Modified Booth algorithm is one of the most popular algorithms used to reduce the number of partial products. In order to achieve speed improvements Wallace Tree algorithm is used which reduces the number of sequential adding stages. The combination of Modified Booth algorithm and Wallace Tree technique can provide advantage of both algorithms in one multiplier.

In the following sections, four multipliers namely, Array multiplier, Modified Booth multiplier, Wallace tree multiplier and modified Booth-Wallace tree multiplier are compared based on various performance measures.

2. ARRAY MULTIPLIER

Array multiplier has a regular structure. These multiplier carries multiplications by the standard add and shift operation using "add and shift" algorithm. The partial products are generated by multiplying the multiplicand with each of the multiplier bit. These partial products are shifted depending on their bit orders and are finally added at the last stage. The numbers of partial products generated is equal to the number of multiplier bits. The addition of these partial products can be done using N-1 normal carry propagate adders where N is the length of the multiplier.

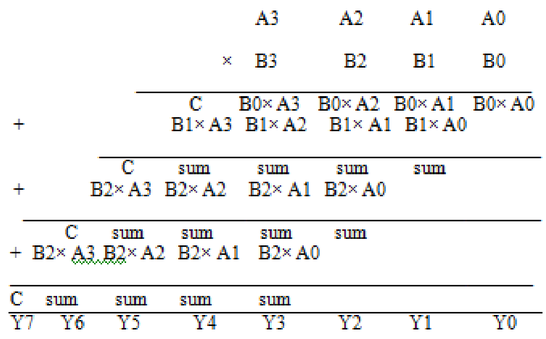
A3 A2 A1 A0
× B3 B2 B1 B0
C B0×A3 B0×A2 B0×A1 B0×A0
+ B1×A3 B1×A2 B1×A1 B1×A0
C sum sum sum sum
+ B2×A3 B2×A2 B2×A1 B2×A0
C sum sum sum sum
+ B2×A3 B2×A2 B2×A1 B2×A0
C sum sum sum sum
Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0

Figure 1.Array Multiplication structure

The multiplicand and multiplier can be either positive or negative and to accommodate this positive and negative value, sign bit extension need to be done at an appropriate stage. The 2's complement of partial products that are added in carry save adder should be of equal bit length in each adder stage. Therefore, the sign bits of the partial product and the sum and carry signals of each adder row are extended up to the most significant sign bit of the number with the largest absolute value to be added in this stage. The sign bit extension will result in higher capacitive load and will result in more power consumption and due to this increase in capacitive load it will lead to reducing the speed of operation.

The block diagram of a 32-bit array multiplier is drawn below

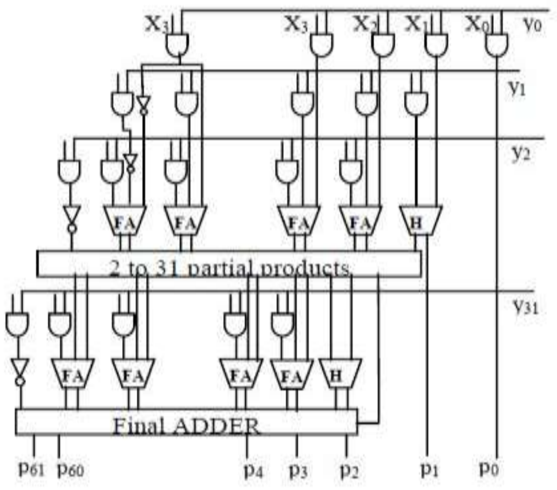

Figure 2. 32-bit Array Multiplier

3. MODIFIED BOOTH MULTIPLIER

Modified booth multiplier uses a powerful algorithm for multiplying signed-number, which is an algorithm that treats both positive and negative numbers uniformly. Modified Booth multiplier design uses an improved modified Booth encoder and selector technique to reduce and rearrange the partial products. This reduces the gate count and improves the performance of the multiplier. The Booth encoder uses Radix-4 or Radix-8 algorithm to encode the multiplier bits. The partial product generator generates partial product from the multiplicand and the encoded multiplier. The partial products are then added by partial product reduction tree (PPRT) without carry propagation. Finally, the summed results obtained from PPRT stage are added using Carry Propagate Adder (CPA).The block diagram of Modified Booth multiplier is shown below:

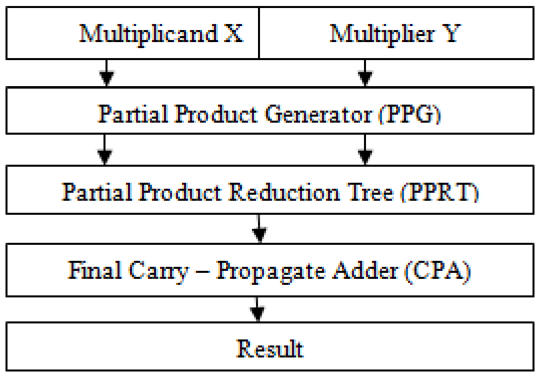
Multiplicand X
Multiplier Y
Partial Product Generator (PPG)
Partial Product Reduction Tree (PPRT)
Final Carry – Propagate Adder (CPA)
Result

Figure 3.Block Diagram of Modified Booth Multiplier

3.1. Radix-4 Booth Algorithm

Booth algorithm is a powerful algorithm used for multiplying signed number, which treats both positive and negative numbers uniformly. A k-bit binary number can be interpreted as a k/2-digit Radix-4 number, a k/3-digit Radix-8 number and so on, that deals with more than one multiplier bits in each cycle for using high radix multiplication. The major disadvantage of Radix-2 algorithm was that the multiplication process requires n shifts and an average of n/2 additions for an n bit multiplier. The variable number of shift and add operations makes it inconvenient for designing parallel multipliers. Also the Radix-2 algorithm is inefficient when there are isolated 1's. The Radix-4 modified Booth algorithm overcomes all these disadvantages of Radix-2 algorithm. The modified Radix-4 Booth algorithm has been widely used when the operands are equal to or greater than 16 bits. In this algorithm, the two's complement multiplier is encoded in order to reduce the number of partial products that needs to be added to n/2. The multiplier, Y can be written in two's complement form as:

$$Y = -Y_{n-1}2^{n-1} + \sum_i Y_i 2^i \;\; ; 0 \leq i \leq n-2$$

It can be also written as

$$Y = \sum_i (-2Y_{2i+1} + Y_{2i} + Y_{2i-1})\, 2^{2i}; \; 0 \leq i \leq n-2$$

The encoding of the signed multiplier Y, using the Radix-4 Booth algorithm is shown below in Table I. The Radix-4 Booth recoding method encodes the multiplier bits into [-2, 2]. The multiplier bits are considered in blocks of three, such that each block overlaps the previous block by one bit. Examining the multiplier with the least significant bit makes it advantageous. The overlapping of block is done so that we know what has happened in the last block, as the most significant bit of the block acts like a sign bit.

TABLE I
Radix-4 recoding

Multiplier Bits			**Recorded Operation on Miltiplicand,X**
Y_{2i+1}	**Y_{2i}**	**Y_{2i-1}**	
0	0	0	0X
0	0	1	+X
0	1	0	+X
0	1	1	+2X
1	0	0	-2X
1	0	1	-X
1	1	0	-X
1	1	1	0X

3.2. Radix-8 Booth Algorithm

Radix-8 Booth recoding applies the same algorithm as that of Radix-4, here instead of triplets, that is, blocks of three, blocks of four are used. Each quartet is encoded as a signed digit using Table II.

TABLE II

Radix-8 recoding

Multiplier Bits				Recorded Operation on Multiplicand, X
Y_{i+2}	Y_{i+1}	Y_i	Y_{i-1}	
0	0	0	0	0X
0	0	0	1	+X
0	0	1	0	+X
0	0	1	1	+2X
0	1	0	0	+2X
0	1	0	1	+3X
0	1	1	0	+3X
0	1	1	1	+4X
1	0	0	0	-4X
1	0	0	1	-3X
1	0	1	0	-3X
1	0	1	1	-2X
1	1	0	0	-2X
1	1	0	1	-X
1	1	1	0	-X
1	1	1	1	0X

4. WALLACE TREE MULTIPLIER

In this architecture, all the bits of all the partial products in each column are added together to a set of counters in parallel without propagating the carries using carry save adders. Another set of counters reduces this new matrix until a two row matrix is generated. A fast adder, that is, carry look ahead adder is at the end to produces the final result which makes the multiplication process fast due to acceleration in accumulation of partial products because of the use of these adders.

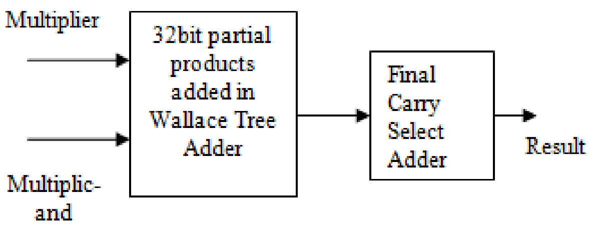

Figure 4.Block diagram of Wallace tree Multiplier

5. MODIFIED BOOTH-WALLACE TREE MULTIPLIER

The process of Multiplication includes two steps, that is, first the generation of partial products and second is their accumulation. The speed of multiplication can be increased either by reducing the number of partial products and/or by accelerating the accumulation of partial products. Booth recoding with Wallace tree adders has been often used in fast multipliers. In the above sections Modified Booth multiplier and Wallace Tree multiplier architectures have been discussed in detail. In this section, a combined Modified Booth-Wallace tree multiplier is explained and its performance is observed with respect to the other multipliers. Modified Booth algorithm provides better area performance and Wallace tree benefits with reduced delay. The block diagram of this multiplier consists of four major modules: a Booth encoder, partial product generator, Wallace tree and the final carry look-ahead adder. The Booth encoder encodes the multiplier bits using Radix-4 or Radix-8 algorithm. Based on the multiplicand and the encoded multiplier, the partial products are generated by using partial product generator. For large multipliers of 32 bits, the performance of modified Booth algorithm is limited. So Booth recoding together with Wallace tree structures have been used in this Modified Booth-Wallace tree multiplier. The partial products generated from the partial product generator are supplied to Wallace Tree and added appropriately. The results are then finally added using a Carry Look-ahead Adder (CLA) to get the final product.

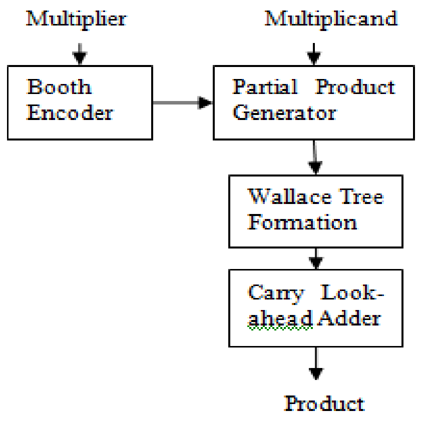

Figure 5.Block Diagram of Modified Booth-Wallace Tree Multiplier

The Modified Booth algorithm reduces the number of partial products to half of number of multiplier bits if Radix-4 booth algorithm is used and reduces to one-third of the number of multiplier bits if Radix-8 booth algorithm is used.Modified Booth algorithm is in order to save chip area rather than reducing delay time and by using Wallace tree operation time is reduced because it uses Carry Save Adders (CSA) for fast accumulation of the partial products. To further increase the speed of operation, carry-look-ahead (CLA) adder is used as the final adder. By using Wallace tree multiplier,power consumtion is also reduced when compared with other multipliers.

6. CONCLUSION

In this section, four multipliers discussed above are compared based on various performance measures. The basic Array multiplier is the simplest of all multipliers in its circuit complexity and occupies small area but these multipliers multiplies at a low speed using the basic add and shift algorithm. Based on the above study, it can be concluded that of all the multipliers, Modified Booth Wallace tree multiplier is the fastest as it takes the advantage of both, Modified Booth multiplier wherein the no of partial products are reduced by half or one third of the multiplier bits based on the algorithm it uses ,that is, Radix-4 or Radix-8 and Wallace tree multiplier increases speed of accumulation because of the use of carry-save adders but it has the drawback of complex circuitry that occupies the largest area. Thus a multiplier should be selected depending of the performance measures and the nature of applications. Based on the discussion in the above sections, comparison of various multipliers is shown in table below:-

TABLE III

Comparison of multipliers based on various parameters

Multiplier Types	**Speed**	**Circuit Complexity**	**Area**
Array	Low	Simple	Small
Modified Booth	High	Complex	Medium
Wallace tree	Higher	Medium	Large
Modified Booth-Wallace tree	Highest	More Complex	Largest

7. ACKNOWLEDGEMENT

We are thankful to our guide Assistant Professor Ajit Saraf (Pillai's Institute of Information Technology, India) for the helpful discussions and valuable guidances

8. REFERENCES

[1] Aparna.P.R and Nisha Thomas, "Design and Implementation of a High Performance Multiplier using HDL".

[2] Lakshmanan, Masuri Othman and Mohamad Alauddin Mohd.Ali, "High Performance Parallel Multiplier using Wallace-Booth Algorithm",Semiconductor Electronics, IEEE International Conference , pp.: 433436,Dec.2002.

[3] Prasanna Raj P, Rao, Ravi, "VLSI Design and Analysis of Multipliersfor Low Power", Intelligent Information Hiding and Multimedia Signal Processing, Fifth International Conference, pp.: 1354-1357, Sept. 2009

[4] Lakshmanan, Masuri Othman and Mohamad Alauddin Mohd.Ali, "High Performance Parallel Multiplier using Wallace-Booth Algorithm",Semiconductor Electronics, IEEE International Conference , pp.: 433436, Dec.2002.

[5] C.S.Wallace,"A Suggestion for a Fast Multiplier", ElectronicComputers, IEEE Transactions, vol.13, Page(s): 14-17, Feb. 1964

[6] Hussin R et al , "An Efficient Modified Booth Multiplier Architecture",IEEE International Conference, pp.:1-4, 2008.

[7] L.P. Rubinfield, A Proof **of** the modified Booth algorithm for multiplication, IEEE Trans on Computers C-24 (Oct-1975) 1014-1015.

[8] A.D.Booth, **"A** signed binary multiplication technique",

[9] V.G OMobdzija, "A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach", *IEEE Transactions* ***on*** *Computers,* Vol. 45, No, 3, pp.294-305, March 1996.