Review of Booth Algorithm for Design of Multiplier

N.VEDA KUMAR, THEEGALA DHIVYA
Assistant Professor, M.TECH STUDENT
Dept of ECE,Megha Institute of Engineering & Technology For womens,Edulabad,Ghatkesar mandal,RangaReddy Dist,Telangana ,India

Abstract—Now a day's many of technologies handles low power consumption to meet the requirements of various outboard applications. In these applications, a multiplier is a fundamental arithmetic unit and used in a great extent in circuits. Multipliers are key components of many high performance systems such as FIR filters, Microprocessor, digital signal processors etc. Booth algorithm is used for design of multiplier but it suffers from some limitations like number of the partial products increases, so area and time delay also increases. To cut down the number of Calculation steps for the partial products, Modified Booth Algorithm [MBA] have been employed. [1] In this paper various techniques and algorithms are analyzed for design of multiplier in terms of delay, area and power consumption.

Keywords— Booth multiplier, Reversible logic gate, Multiplier and-Accumulator (MAC), Modified booth algorithm.

I.INTRODUCTION

The Continuous advances of the microelectronic technologies make better use of input energy, to encode the data more efficiently, to transmit the information faster and reliable, etc. So in these applications, a multiplier is a fundamental arithmetic unit and used in a great extent in circuits. Multiplier is a widely component for digital signal processing (DSP) applications. The multiplier is implemented with a large of area in very-large-scale integration (VLSI) designs. The fastness of multiplication and addition arithmetic's decides the execution speed and performance of the total calculation. Because the multiplier needs the longest delay within the basic operational blocks in digital systems, the critical path is evaluated by multiplier in general. The multiplier is implemented with a large of area in very-large-scale integration (VLSI) designs A multiplier uses Booth's algorithm and array of full adders (FAs), or Wallace tree instead of the array of FAs. i.e., this multiplier mainly consists of the three parts: i.Booth encoder, ii.a tree to compress the partial products such as Wallace tree, and iii. Adder. In fast digital multiplier design, modified Booth encoding algorithm is an efficient way to reduce the number of partial products by grouping consecutive bits in one of the two operands to form the signed multiples. [1]

It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. Booth algorithm is a method that will reduce the number of multiplicand multiples.

II.BASIC BOOTH MULTIPLIER

Booth Multiplier reduces number of iteration step to perform the multiplication as compare to Conventional steps. The architecture comprises of four parts: Complement Generator, Booth Encoder, Partial product and Carry Look-ahead Adder as shown in fig 1.

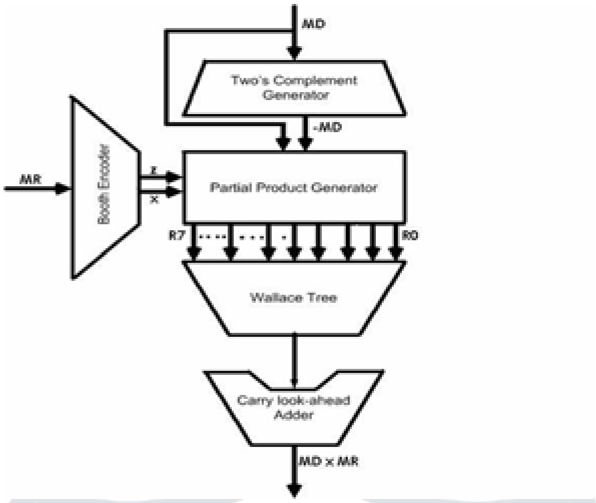


Figure 1: Architecture of Booth Multiplier [Google images]

Booth algorithm scans the multiplier operand and skips chain of this algorithm can reduce the number of additions required to produce the result compared to Conventional multiplication algorithm, where each bit of the multiplier is multiplied with the multiplicand and the partial products are aligned and added together. More interestingly the number of additions is data dependent, which makes this algorithm.

III. REVIEW OF BOOTH ALGOTIHM (RADIX 2)

The Booth algorithm was invented by A. D. Booth forms the base of Signed number multiplication algorithms that are simple to implement at the hardware level, and that have the potential to speed up signed multiplication considerably. Booth's algorithm is based upon recoding the multiplier, y, to a recoded, value, z, leaving the multiplicand, x, unchanged. In Booth recoding, each digit of the multiplier can assume negative as well as positive and zero values. There is a special notation, called signed digit (SD) encoding, to express these signed digits. In SD encoding +1 and 0 are expressed as 1 and 0, but -1 is expressed as 1[2].

The value of a 2s complement integer was defined a by equation 1.

y= -ym-1 2m-1 + 2i (1)

This equation says that in order to get the value of a signed 2's complement number, multiply the m – ith digit by -2`-1, and multiply each remaining digit i by +2g.

For implementing booth algorithm most important step is booth recoding. By booth recoding we can replace string of 1s by 0s.

Hence if this number were to be used as the Multiplier in a multiplication, we could replace five additions by one addition and one subtraction.

The Booth recoding procedure, then, is as follows:

1. Working from lsb to msb, replace each 0 digit of the original number with a 0 in the recoded number until a 1 is encountered.

2. When a 1 is encountered, insert a 1 at that position in the recoded number, and skip over any succeeding I's until a 0 is encountered.

3. Replace that 0 with a 1 and continue.

modified Booth multiplier is shown in fig 3.

Booth Algorithm:

1. Add 0 to right of LSB of multiplier and look at rightmost of multiplier to make pairing of 2 bits from right to left and mark corresponding multiplier value as shown in fig.1

2. 00 or 11: do nothing

3. 01: Marks the end of beginning of a string of 1s Subtract multiplicand from partial product.

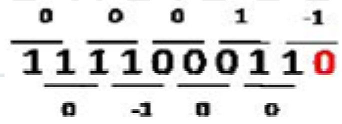


Figure 2: Bit Pairing as per booth encoding [2]

One of the solutions realizing high speed multipliers is to enhance parallelism which helps in decreasing the number of subsequent calculation stages. The Original version of Booth's multiplier (Radix-2) had two drawbacks.

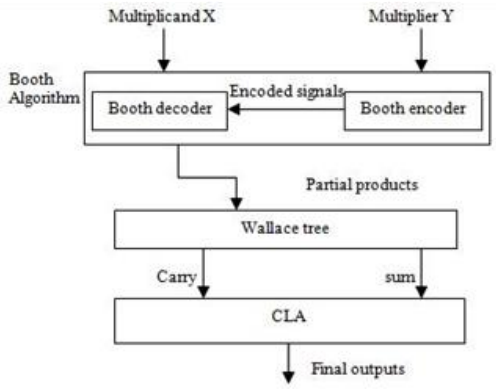
1. The number of add/subtract operations became variable and hence became inconvenient while designing Parallel multipliers.

2. The Algorithm becomes inefficient when there are isolated 1s.

These problems are overcome by using Radix 4 Booth'sAlgorithm which can scan strings of three.

IV. REVIEW OF MODIFIED BOOTH ALGORITHM (RADIX 4)

The modified-Booth algorithm is extensively used for high-speed multiplier circuits. Architecture of the modified Booth multiplier is shown in fig 3.



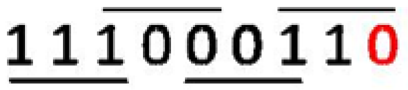


Figure 3.Architecture of the modified Booth multiplierFigure 4. Bit Pairing as per booth encoding [2]

It is possible to reduce the number of partial products by half, by using this technique of Radix 4 Booth encoding. The basic idea is that, instead of shifting and adding for every column of multiplier term and multiplying by 1 or 0, we only take every second column, and multiply by +_1,+_2,or 0, to obtain the same results.

Radix 4 booth encoder performs the process of encoding the multiplicand based on multiplier bits. It will compare 3 bits at a time with overlapping technique. Grouping starts from the LSB, and the first block only uses two bits of multiplier and assumes a zero for the third bit as shown by figure 4.

Radix-4 Booth algorithm is given below:

(1) Extend the sign bit 1 position if necessary to ensure that n is even.

(2) Append a 0 to the right of the LSB of the multiplier.

(3) According to the value of each vector, each Partial Product will be 0, +y, –y, +2y or –2y.

The negative values of y are made by taking the 2’s complement. The negative values of y are made by taking the 2’s complement and Carry-look-ahead (CLA) fast adders are used for addition in this paper. The multiplication of y is done by shifting y by one bit to the left. Thus, in any case, in designing n-bit parallel multipliers, only n/2 partial products are generated [2].The advantage of this method is the halving of the number of partial products. This is important in circuit design as it relates to the propagation delay in the running of the circuit, and the complexity and power consumptionof its implementation**.**

V.COMPARISON OF RADIX2 AND RADIX4

Here is the comparison of two booth multipliers and their characteristics in terms of multiplication speed, no of computations required, no of hardware. From results it is shown that radix 4 booth multipliers is better than Radix-2 booth multipliers and also Radix -4 multiplier computation speed is higher than Radix-2 and Circuit Complexity is also less as compared to it. Comparison is shown below in table1 in terms of number of slices, no of inputs and outputs, lathes etc.

Table 1
Comparison of Radix2 and Radix4 [2]

Device Utilization Summary	**Radix 2**	**Radix 4**
Number of Slices	397	71
mber of 4 input LUTs	184	100
Number of bonded INPUT	16	16
Number of bonded OUTPUT	16	16
Macro Statistics		
# Latches	24	12
8-bit Latch	24	12
# Xors	71	23
1-bit xor2	64	21
8-bit xor2	7	2
Timing Summary		
Minimum period:	5.454ns	4.750ns
Minimum input arrival time before clock	7.936ns	4.014ns
Maximum output required time after clock	6.216ns	6.205ns

Table 2
Comparison of Modified booth algorithm with and without reversible gate logic [4]

Parameter	**MBE**	**MBE with reversible gate logic**
No. of Slices used	178 out of 4656 (3%)	162 out of 4656 (3%)
No. of LUTs	355 out of 9312(3%)	294 out of 9312 (3%)
Total delay	34.535ns	30.38ns

REFERENCES

[1] K.Nagarjun, S.Srinivas, “A New Design of Multiplier using Modified Booth Algorithm and Reversible Gate Logic,”International Journal of Computer Applications Technology and Research, Vol.2, Issue.6, pp.743-747, 2013

[2] SukhmeetKaur, Suman and ManpreetSignhManna,“Implementation of Modified Booth Algorithm (Radix 4) and itsComparison with Booth Algorithm (Radix-2)” Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 3, 2013, pp. 683-690.

[3] NishatBano,” VLSI Design of Low Power Booth Multiplier”International Journal of Scientific & Engineering Research, Volume 3, Issue 2, 2012 ISSN 2229-5518.

[4] K.V.Ganesh, T.Sudha Rani, P.N.VenkateswaraRao, K.Venkatesh,”Constructing a low power multiplier using Modified BoothEncoding Algorithm in redundant binary number system”International Journal of Engineering Research and Applications Vol. 2, Issue 3, May-Jun 2012, pp.2734-2740.

[5] Soojin Kim and Kyeongsoon Cho, “Design of high-speed modified booth multipliers operating at GHz range”. World academy of science, Engineering and Technology 61, 2010.

[6] 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.

[7] G.Jaya Prada, N.C. Pant, “Design and Verification of Faster Multiplier”, international journal of engineering Research andApplications(IJERA), Vol. 1, Issue 3, pp.683-686.

[8] RazaidiHussin, Ali Yeon Md. Shakaff, NorinaIdris, ZalimanSauli, RizalafandeChe Ismail and AfzanKamarudin,”An EfficientModified Booth Multiplier Architecture” International Conference on Electronic Design, 2008, Penang, Malaysia.

[9] C.N. Marimuthu, P. Thangaraj, “Low power high performance multiplier”, ICGST-PDCS, Volume 8, Issue 1, December 2008.

[10] O. L. MacSorley, “High speed arithmetic in binary computers,”Proc.IRE, Vol.4, Issue.2, pp. 67–91, February 1999.

[11] http: //virtual-labs.ac.in/labs/cse10/booth.html.

[12] http://www.academia.edu/5086115/ VLSI Design of Low Power Booth Multiplier.