International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Review on Advances in VLSI Multipliers

Amol Baile[1], Priti Ingle[2], Prof. Shubhangi Joshi[3]

Electronic And Communication Department, MIT ADT University, India

Abstract:
This abstract explores recent advancements in multiplier design, addressing the critical factors of speed, power consumption, and layout regularity. The emphasis is on achieving optimal trade-offs to meet the increasing demands of high-performance digital systems. As digital signal processing (DSP) systems continue to evolve, the reliance on hardware multiplication becomes paramount for achieving high data throughput. The presented advancements in multiplier design contribute to the development of compact, high-speed, and low-power Very Large-Scale Integration (VLSI) implementations, catering to the diverse needs of modern digital systems.

Keywords: High speed, compact, power consumption.

Introduction

Very Large Scale Integrated Circuit (VLSI) in this low power very large scale integration circuit is an important criteria for designing an energy efficient design for its optimal performance and the compact device design. Multiplier plays an important role for planning the economical energy efficiencies for a low power design. Multiplier has important role in DIP, microprocessor, and microcomputer application. In this paper we are going to review the different Multipliers that are Array multiplier, Booth multiplier, Modified booth multiplier and many more.

All applications need speed multipliers. When comparing different balances, we find that choosing the ri ght balance for each application is always a balance between speed, power and space. The array multipli er has the largest latency, least power consumption, and largest area, while the cabinetcoded Allace tree multiplier has the smallest latency and largest area. One of the Constraints in the above equation are that there are no parts of the product. The cabinet merging algorithm overcomes this limitation and thus impr oves performance. In this review, the booth algorithm and its variant which outperform the several multi pliers will be discussed. The detail comparison of multipliers and their performance matrix in term of del ay, area and power consumption will be provided, so that the most appropriate multiplier can be chosen f or application.

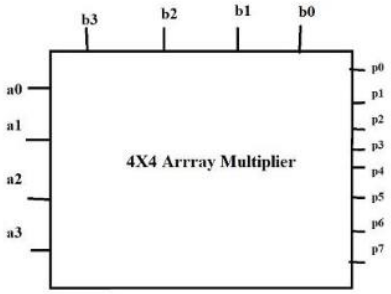
Types of Multipliers

Array Multiplier

The array multiplier is of linear combination; An array multiplier is a digital combination that uses an a rray of full adder and half adder to multiply two binary numbers. This multiplier makes the equation usin g standard addition and substitution algorithms. Partial multiplication occurs by multiplying a number by each digit of the same number. The partial result is then modified in bit order and finally added to the fin al stage.

It needs to have more gates, so the area increases. As the area increases, the delay also increases, which i

ncreases the complexity throught of the circuit. Array multipliers have greater latency than Wallace tree multipliers. The worstcase delay of a string multiplier is (2n + 1) . It performs equations with addition an d substitution algorithms. N multipliers require n-1 adders. The number of half-produced items is equal to the number of duplicate items.

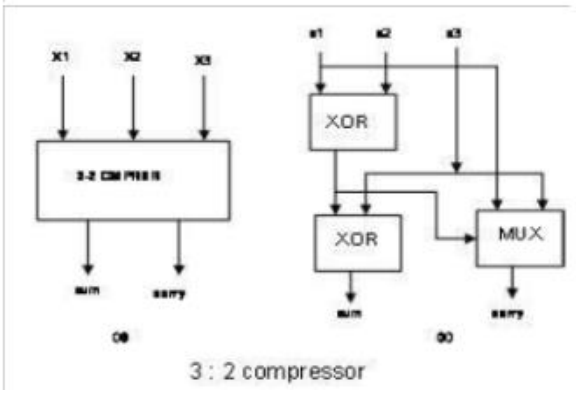


WALLACE TREE MULTIPLIER

The multiplier was first discovered in 1964 by Australian computer scientist Chris Wallace. In the Walla cetree multiplier, partial sum adders are used in a tree like fashion.

By using halfsum adders, the number of critical paths and adders is reduced. This multiplier reduces har dware usage and keeps it as a large multiplier, so latency tends to zero. The Wallace tree multiplier lessens the variety of partial merchandise and use carry pick out adder for the addition of partial merchandise. Wallace tree is an irregular in shape in which casual implementation does no longer deliver the systematic method for the compressor interconnection, still it's far an efficient method of adding partial product in parallel. This model uses a three-step process to combine two numbers. Wallace tree multipliers come in two different designs. The forme r aims to use only half and full adders, while the latter uses a complex carryskip adder (CSA). In the first step, each bit of the product is multiplied by one bit of the multiplier to form a portion of the product.

The speed, area and power consumption of the multiplier will be directly proportional to the efficiency o f the compressor. Compressor (3:2) is used to make the whole equation faster. Using the Wallace tree of the 3:2 compressor, our partial output is sent to the full adder, called our Wallace tree circuit concept, an d the output of the signal is used for all stages next to the same number of elements. . collector, the entire collector is slightly high.



3 : 2 compressor

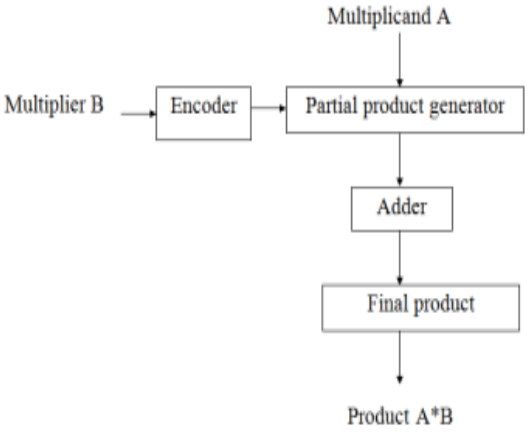
The schematic diagram of a 4*4 Wallace tree multiplier is as below

				A_3 B_3	A_2 B_2	A_1 B_1	A_0 B_0
				P_{03}	P_{02}	P_{01}	P_{00}
			P_{13}	P_{12}	P_{11}	P_{10}	
		P_{23}	P_{22}	P_{21}	P_{20}		
	P_{33}	P_{32}	P_{31}	P_{30}			
	P_{33}	P_{23}	S_{04}	S_{03}	S_{02}	S_{01}	P_{00}
		C_{04}	C_{03}	C_{02}	C_{01}		
	P_{33}	P_{32}	P_{31}	P_{30}			
	P_{33}	S_{14}	S_{13}	S_{12}	S_{11}	S_{01}	P_{00}
	C_{14}	C_{13}	C_{12}	C_{11}			
C_{out}	S_{24}	S_{23}	S_{22}	S_{21}	S_{11}	S_{01}	P_{00}

BOOTH MULTIPLIER

The algorithm was developed by Andrew Donald Booth in 1950. While doing research in crystallograph y, Booth used a desk calculator that could be converted faster than addition and developed a method to s peed it up. It holds both signed and unsigned numbers. It is a very popular signed integer multiplication algorithm that calculates positive and negative integers. An algorithm that uses the addition of 2 representation of signed binary numbers for multiplication.

Each equality bit results in a set of equal parts that must be added to the partials. In this case, the delay o f the multiplier is determined only by the number of additions to be made. Performance will be better if t here is a way to reduce the number of insertions. It is a standard process used in wafer production a nd provides significant improvements in equivalent length. The Booth algorithm flow chart is as below



Booth algorithm is a method that will reduce the number of multiplicand multiples.

MODIFIED BOOTH MULTIPLIER:

Wen Chang Yeh and Chein Wei Jen proposed a new design based on the new development of modified coding scheme (MBE) which tries to improve the performance of the MBE model. They also created a n ew aggregator for last insertion that is faster and up to 25% faster, but they also said that the use of the n ew update table should be better considered when there is different logic.

It reduces the number of factors and the number of partial products.

It groups us together successively at a time; where the 2 bits we leave are the current bit and the third bit is a bit carried over from the previous bit pair. For longer operations, array multipliers are faster than arr ay multipliers because the execution time of the operation is proportional to the logarithm of the length o f the operand. The main advantage of this challenge is that addition can be skipped if consecutive items i n the equation are the same. Quantities of some products were reduced by half using booth coding techn

ology.

At a certain multiplier, three groups of three products are taken into account, starting from LSB products , the first group consists of two products, the next three groups of three products are taken into account, one will overlap with the previous group. Therefore, the multiplier group will produce the following pro ducts of these five products, denoted as -2, -1, 0, +1 and +2

Block	Recoded digits	Operation
000	0	0
001	+1	+1
010	+1	+1
011	+2	+2
100	-2	-2
101	-1	-1
110	-1	-1
111	0	0

Multiplier types	Speed	Power Consumption	Circuit complexity	Delay time (NS)
Array multiplier	less	highest	easy	72.986
Wallace tree multiplier	higher	Moderate	moderate	53.198
Booth multiplier	high	low	complex	55.70
Modified booth Multi	highest	low	Most complex	54.50
Shift and add multiplier	less	higher	less	222

Diagram of Modified Booth algorithm is shown. Partial Product Generator [PPG] produces partial products by connecting and matching n umbers. [6] The partially obtained PPG output is then added to the PPRT without propagation. Fi nally, the results obtained during PPRT are added to the carrier propagation sequence [CPA].

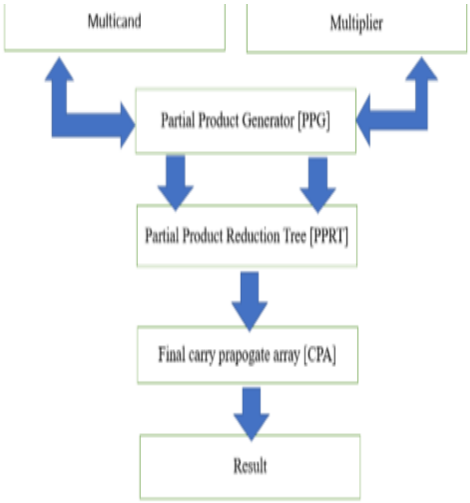


Figure 6 Block diagram of Modified Booth Algorithm

SHIFT AND ADD MULTIPLIER

The figure below shows the general design of the shift and addition multiplier for a 32-bit multiplier. The value of the balance

is added and written according to the value of the LSB multiplication of the balance. In each clock cycle, the equation is shifted to the right by one and its value is checked.
If it is 0,only the function changes. If the value is 1, the multiplier is added to the accumulator and shifte d one position to the right. Once allmultiplier items have been tested the items are entered into the accu mulator. The size of the accumulator is 2N(M + N) and contains the first N LSB multipliers. The delay i s up to N cycles. This circuit has many advantages over non-synchronous circuits.

COMPRASION TABLE 1.

CONCLUSION

Through these studies, we have learned that there is often a tradeoff
between the effectiveness of multipliers.
Boothmultiplier is the best in terms of speed compared to other multipliers. The delay of the Wallace tre e multiplier is smaller than other multipliers. Various types of multipliers are examined, including perfor mance
parameter andcomplexity. From a simple perspective, the performance of array multipliers is better than all multipliers, but the speed is the slowest. If we consider the power consumption of the multiplier, whic h is the most important parameter in portable devices, modified Booth multiplier to perform well of all the discussed multipliers. The area consumption of the array multiplier is small, while that of Wallace tree multiplier is large, For booth multiplier it is moderate, for Modified booth multiplier it is the largest & for shift and sum multiplier it is comparatively less.

REFRENCES

1. K. N. Singh and H. Tarunkumar, "A review on various multipliers designs in VLSI," Annual IEEE India Conference, pp 1-4, 2015
2. B. Lamba and A. Sharma, "A review paper on different multipliers based on their different performance parameter", 2nd International Conference on Inventive Systems and Control, pp 324-327, 2018
3. Saokar, S.S.; Banakar, R. M.; Siddamal, S., "High speed signed multiplier for Digital Signal Processing applications," IEEE International Conference on Signal Processing, Computing and Control (ISPCC), vol.15, no.17, pp.1-6, March 2012. [4]. C.S. Wallace, "A Suggestion for a Fast Multiplier", IEEE Transaction On Electronic and Computer, Vol.EC-13, PP. 14-17, February 1964.
4. Moises E. Robinson and Earl Swartzlander, Jr., "A Reduction Scheme to Optimize the Wallace Multiplier", IEEE Proc. Int. Conf., Compute. Design: VLSI in Compute. and Processors, pp. 122-127, Oct. 1998. 3 [6] Ron S. Waters and Earl E. Swartzlander, Jr., "A Reduced Complexity Wallace Multiplier Reduction", IEEE Trans., pp. 1134-1137, Aug., 2010.
5. Savita Nair, Ajit Saraf, "A Review Paper on Comparison of Multipliers based on Performance Parameters", ICAST, 2014.
6. Soniya, Suresh Kumar, "A Review of Different Type of Multipliers and Multiplier-Accumulator Unit", IJETTCS, Vol.2, Issue 4, July-August 2013. [8]. http://www.dauniv.ac.in/downloads/CArch_PP Ts/CompArchCh03L06ArrayMult.pdf