© 2018 IJSRSET | Volume 4 | Issue 4 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Review of Multipliers

R. Senthil Ganesh, K. Hemamalini, V. Indhu, S. Kamala Prabha
Electronics and Communication Engineering, Info Institute of Engineering, Coimbatore, Tamil Nadu, India

ABSTRACT

Multiplier is one of the basic functional unit is DSP. The methods for verifying multipliers based on symbolic function representation. Bit-level verification is performed. Multipliers play an important role in the high performance digital systems. Multipliers design considerations include the following low power consumption, high speed, regularity of layout. It is suitable for various compact high speeds, low power VLSI implementations. Booth multiplier is parallel multiplier that uses carry look ahead algorithm. To reduce latency and improve speed the booth multiplier is proposed in it. The best architecture for multiplier with respect to power and delay characteristics. To achieve high data throughput the DSP systems relay on hardware multiplication.

Keywords : Digital Signal Processing (DSP), Finite Impulse Response (FIR), Arithmetic Logical Unit (ALU).

I. INTRODUCTION

The simple common multiplication method is shift and add algorithm [2]. The key components of multipliers are high performance systems such as FIR filters, Microprocessors, DSP, ALU, etc., generally the slowest element in the system is multiplier. It consumes more power, area and latency. In DSP applications, fast multipliers are essential. Therefore high speed multiplier is much desired. The instruction cycle time of a DSP chip is determined by the multiplication in dominant factor [2]. Multiplier concept is performed by Number of addition of operation. Number of partial products added in parallel multipliers is main parameter that determines the performance of multiplier [1]. Arithmetic unit is the multipliers main block and it has Low Power consumption, high speed and each logic has its own advantages in terms of speed and power. We introduce the difference types of multiplier are Wallace tree multiplier, booth multiplier, Vedic multiplier, array multiplier, dada multiplier.

To perform a multiplication a single two input adder is used. The M and N bits is the input for basic algorithm [1]. The shift algorithm is a common multiplication. Binary multiplications, long hand decimal multiplication, the addition of shifted versions are involved. In multiplier bits the value and positions are based on multiplicand. Array multiplier is a resulting algorithm.

II. MULTIPLIERS

A. Array Multiplier

The array multiplier is like normal multiplier, it is an efficient layout of combinational multiplier [2]. In this partial products are generated by using two input AND gates [1]. Optimum numbers of components are used in this multiplier. Nxn multiplier requires n (n-2) full adders, n half adders and n^2 AND gates. It requires large number of gates so, area gets increased, because of increased area delay was also increased it also increases circuit complexity and also it's not economical. The delay of array multiplier is greater than the Wallace–tree multiplier. The worst case delay of array multiplier is (2n+1) [2]. It performs multiplication by add and shift algorithm. N multiplier requires n-1 adders [6]. Number of partial products generated is equal to the number of multiplier bits. Capacitive load increases for sign bit

extension then the other signals; this will reduce the speed of the circuit. For high speed array multiplier circuit, delay depends upon the bit size of multiplicand and multiplier [8].

Algorithm:

Step 1: Convert the given multiplicand and multiplier in to binary form and then multiply each bit of the multiplier with each bit of the multiplicand.

Step 2: If the multiplier bit was one then, the multiplication output of multiplier bit one with the multiplicand is same as the multiplicand.

If the multiplier is bit was zero then, the multiplication output of the multiplier bit zero with the multiplicand was zero.

Step 3: Finally add the products using full adders and half adders respectively.

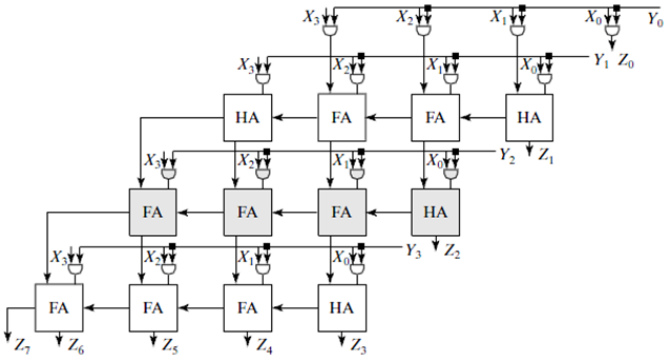

Figure 1. Array Multiplier

Wallace-tree:

This multiplier was first proposed by Australian computer scientist Chris Wallace in 1964. In Wallace-tree multiplier partial sum adders are used in a tree like fashion, by using partial sum adders critical path and number of adders are reduced. This multiplier reduces hardware usage and saving it for large multiplier so delay is reduced, it is almost equal to zero [1]. It is faster than carry save structure especially for large multiplier word lengths [2]. It will provide low power dissipation for all possible input combinations, the bit products are calculated and they are arranged in a row matrix. Sum of the bit products equal to the sum of the row. The addition of rows gives the final result by using fast adders. Three steps are required for the multiplication of two numbers [5].

If the bits are arranged in a tree like fashion, they will look like a tree of carry save adders. Carry save adders is like ripple carry adder, the speed, area and power consumption are directly proportional to the efficiency of the compressor. In this carry is saved instead of sending carry to the next stage. Finally the carry is added to the sum later [6].

Algorithm:

Three steps are involved in the multiplication process are….

Step 1: Multiply each bit of the multiplier with each bit of the multiplicand and produce n^2 results.

Step 2: Reduce the partial products in the row matrix form by using half adders and full adders.

Step 3: Finally add the reduced row of partial products using conventional full adders.

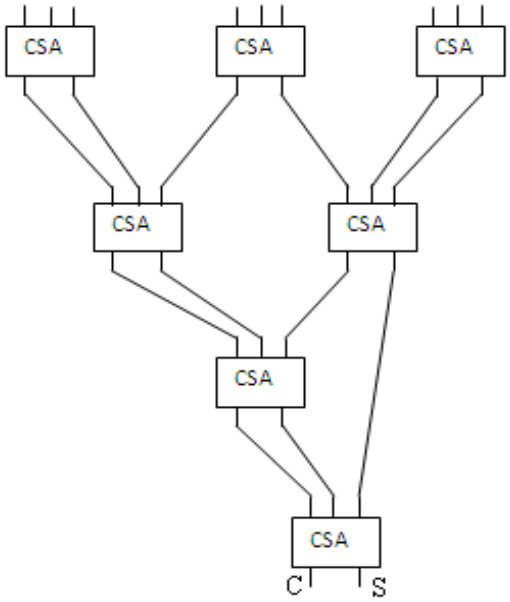

Figure 2. Wallace-tree Multiplier.

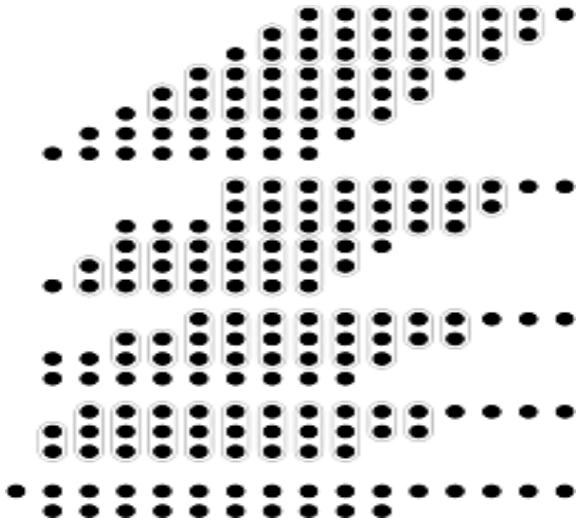

Figure 3. Steps for Wallace-tree Multiplier

Booth Multiplier:

Booth multiplier was first proposed by Andrew Donald Booth in 1951. It will treat both the signed and unsigned numbers equally, but the multiplication of signed numbers should take special care during calculation. In this multiplier, multiplier is recoded using booths recoding table and then only the multiplication is carried out by using booth algorithm. It will reduce the number of multiplicand and the number of partial products. The partial products are reduced by scanning 3 bits at a time, in that three bits 2 bits are present bits and third bit is the carry bit from the previous pair of bits. The main advantage of this multiplier is that if the successive bits are multiplied then addition can be skipped [5]. The delay is determined by the number of additions performed by the multiplier. The sign of the operands stored in the auxiliary circuits will increase the complexity, so it consumes more power [9].

TABLE 1. Truth table for Booth Multiplier

Qn	Qn+1	Recoded Bits	Operation
0	0	0	Shift
0	1	+1	Add X
1	0	-1	Subtract X
1	1	0	Shift

Algorithm:

Step 1: Check whether the multiplier and multiplicand are having negative sign; if so convert that into 2's complement form.

Step 2: Add implied zero to the multiplier and then convert the multiplier into other form using booth recoding table.

Step 3: Now multiply the multiplier with multiplicand.

If the multiplier was zero then the multiplicand multiplied with zero will result in the output as zero.

If the multiplier was +1 then, the result of the multiplicand with the multiplier was same as the multiplicand.

If the multiplier was -1 then, the result of the multiplication was calculated by taking 2's complement of the multiplicand.

Step 4: Finally add all the partial products using full adders and half adders respectively.

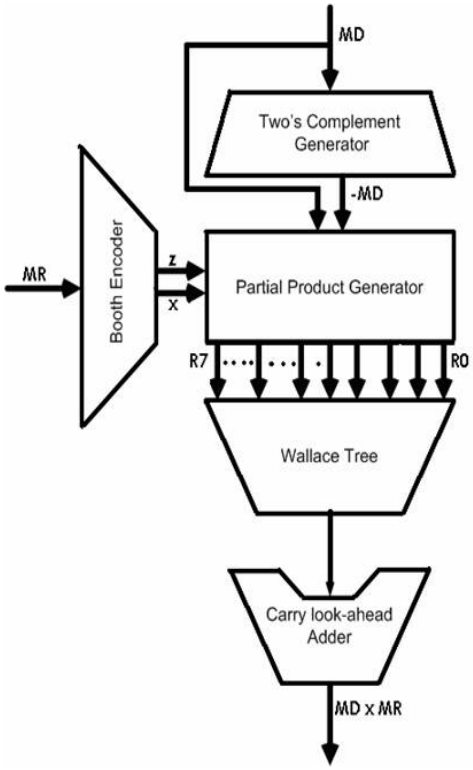

Figure 4. Booth Multiplier

Dadda Multiplier:

It was first proposed by computer scientist Luigi Dadda in 1965. Dadda multiplier is similar to Wallace-tree multiplier [10]. It is faster than array

multiplier and requires less number of gates. The reduction phase is less expensive in this multiplier. Its very speed, to achieve this second step is made more complex than the Wallace –tree multiplier. Delay in each stage is almost same. N by n partial products is produced when multiplying N-bit multiplier. The height of each stage are in the order of 2,3,4,6,9,13,19,28,42,63. N^2-4N+3 full adders and N-1 half adders are used in these multipliers [10]. Number of reduction stages for the implementation of Dadda multipliers areas.

Algorithm:

Step 1: Multiply each bit of the multiplicand by multiplier and produce n^2 partial products.

Step 2: Reduce the partial products to two rows using the half adder and full adder.

Step 3: Finally add all the result using the carry propagate adder.

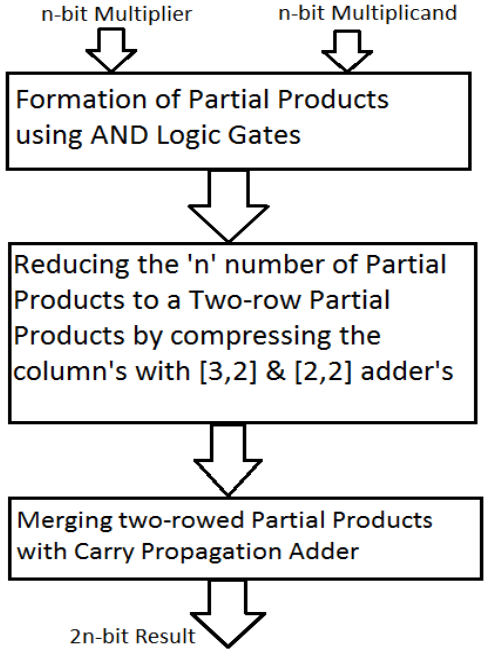

Figure 5. Dadda Multiplier

Vedic Multiplier:

Vedic multiplier is based on the Vedic mathematics sutras in that Urdhva-Triyagbhyam is the most frequently used sutra for the multiplication [3]. Vedic multiplier like general multiplier and its applicable to all cases and the delay in the circuit is reduced, because the partial products are generated parallel and it is added to the previous carry and then final result is calculated. This process of multiplication is same for the multiplication of binary numbers. Its like a general multiplication formula. Delay will get increases when we multiply large numbers [5]. The final product in the Vedic multiplier is similar to the array multiplier [4]. It has high carry propagation delay in case of large numbers. The delay in this multiplier is lesser than array and Booth multiplier.

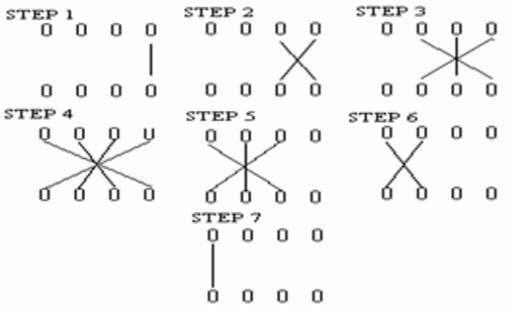

Figure 6. Steps for Vedic Multiplier

Algorithm:

Step 1: Multiply the last bit of the multiplicand with the multiplier and added with the previous carry to get the exact result by using full adders. Initially carry is made zero.

Step 2: Multiply the multiplicand with the other consecutive multipliers bits, if there are more lines at single step then all the result are added and the process is carried out for all the bits.

Step 3: If the last bit of the multiplier is multiplied with the multiplicand then the multiplication process will get finished. The multiplication of the last bit will give the finally carry and the result of the multiplication.

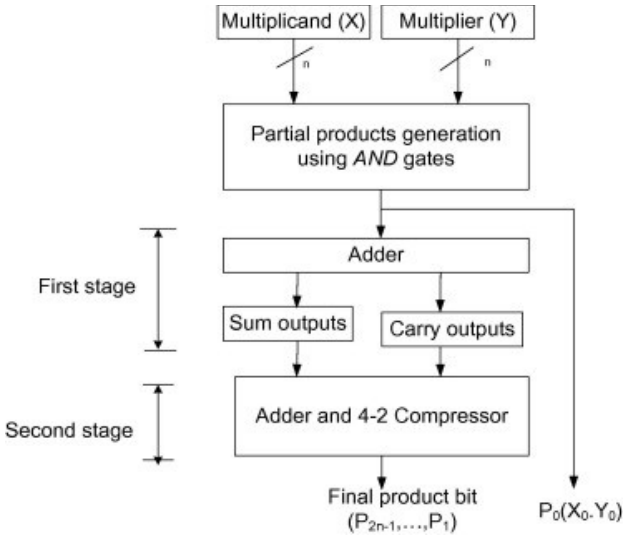

Figure 7. Vedic Multiplier

III. CONCLUSION

From the above discussion of various multipliers such as booth, Vedic, array, Wallace-tree and dada multiplier in that booth multiplier is the best in case of speed when comparing to other multipliers. The delay of Wallace-tree multiplier is less than other multipliers. Vedic multiplier is best in case of the applications such as FFT, Convolution, MAC unit when compared with other multipliers. Booth multiplier will consume more power but it is more efficient comparing to other multipliers.

IV. REFERENCES

[1]. http://shodhganga.inflibnet.ac.in/bitstream/10603/6521/10/10_chapter%205.pdf

[2]. Sumit Vaidya, Deepak Dndekar, "Delay-Power Performance Comparison of Multipliers in VLSI Circuit Design", IJCNC, Vol.2, No.4, July 2010.

[3]. http://www.vedamu.org/veda/1795$vedic_mathematics_methods.pdf

[4]. Eppili Jaya, K.Chitambara Rao, "Power, Area and Delay Comparision of Different Multipliers", IJSETR, Vol.5, Issue 6, June 2016.

[5]. Sumit R. Vaidya, D. R. Dandekar, "Performance Comparison of Multipliers for Power-Speed Trade-off in VLSI Design", ISSN: 1709-5117, ISBN: 978-960-474-162-5.

[6]. Savita Nair, Ajit Saraf, "A Review Paper on Comparison of Multipliers based on Performance Parameters", ICAST, 2014.

[7]. 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_PPTs/CompArchCh03L06ArrayMult.pdf

[9]. https://books.google.co.in/books?id=-9DjuAAACAAJ&dq=booth+multiplier&hl=en&sa=X&ved=0ahUKEwjXiJyOs-vZAhULP48KHbtCAgwQ6AEIKDAA

[10]. Anju.S, M. Saravanan, "High Performance Dadda Multiplier Implemention using High Speed Carry Select Adder", IJARCE, Vol.2, Issue 3, March 2013.

[11]. https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Dadda_multiplier.html

[12]. Kripa Mathew, S. Asha Latha, T. Ravi, E. Logashanmugam, "Design and Anaysis of an Array Multiplier Using an Area Efficient Full Adder Cell in 32nm CMOS Technology, IJES, Vol.2, Issue 3, March 2013.

[13]. Swetha.S, Khobragade, Swapnili P. Karmore, "Low Power VLSI D esign of Modified Booth Multiplier", Recent Trends in Engineering and Technology, Vol.9, July 2013.