

Design of High Performance 64 bit MAC Unit

**G.Pushpa**

M.Tech Student,
Department of ECE,
KITS for Women's, kodad, T.S, India.

**MS.K.Anuradha**

Associate Professor,
Department of ECE,
KITS for Women's, kodad, T.S, India.

ABSTRACT:

A design of high performance 64 bit. Multiplier-and-Accumulator (MAC) is implemented in this paper. MAC unit performs important operation in many of the digital signal processing (DSP) applications. The multiplier is designed using modified Wallace multiplier and the adder is done with carry save adder. The total design is coded with verilog-HDL and the synthesis is done using Cadence RTL complier using typical libraries of TSMC O.18um technology. The total MAC unit operates at 217 MHz. The total power dissipation is 177.732 mW.

Keywords :

Modified Wallace multiplier, Carry saveadder, multiplier and accumulator (MAC)..

1. INTRODUCTION:

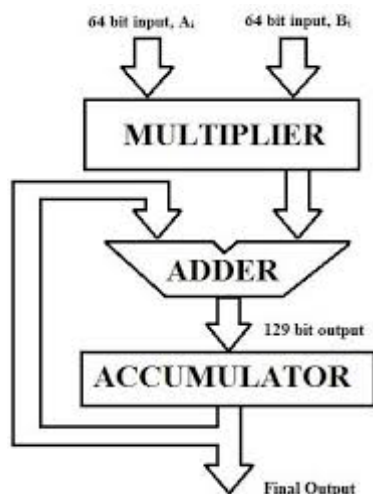
A MAC unit consists of a multiplier and an accumulator containing the sum of the previous successive products. The MAC inputs are obtained from the memory location and given to the multiplier block. Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area-speed constraints have been designed with fully parallel. Hence, with the suitable choice of the type of the multiplier the performance of the MAC unit can be made better.

Hence, in the construction of the MAC unit multiplier plays a vital role so, we have selected those multipliers which exhibit better performance than the previous one implemented in the MAC unit like Vedic Braun, Dadda multiplier and compared the performance with Wallace multiplier which was implemented before in the base paper. This paper is divided into eight sections. In the first section the introduction is discussed. In the second section discuss about MAC operation. The third section details about vedic multiplier is done and fourth section deals with the operation of Dadda multiplier and In the fifth section describes bruau multiplier hence, carry save adder is explained in sixth section and finally obtained results is discussed in seventh and the conclusion is made in the eight section.

2. METHODOLOGY:

MAC Unit is an inevitable component in many digital signal processing (DSP) applications involving multiplications and/or accumulations. MAC unit is used for high performance digital signal processing systems. The DSP applications include filtering, convolution, and inner products. Most of digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transforms (DWT). Because they are basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication and addition arithmetic determines the execution speed and performance of the entire calculation. Multiplication –and -accumulate operations are typical for digital filters. Therefore, the functionality of the MAC unit enables high-speed filtering and other processing typical for DSP applications. Since the MAC unit operates completely independent of the CPU, it can process data separately and thereby reduce CPU load. The application like optical communication systems which is based on DSP ,

require extremely fast processing of huge amount of digital data. The Fast Fourier Transform (FFT) also requires addition and multiplication. 64 bit can handle larger bits and have more memory. A MAC unit consists of a multiplier and an accumulator containing the sum of the previous successive products. The MAC inputs are obtained from the memory location and given to the multiplier block. The design consists of 64 bit modified Wallace multiplier, 128 bit carry save adder and a register. This paper is divided into six sections. In the first section the introduction about MAC unit is discussed. In the second section discuss about the detailed operation of MAC unit. The third and fourth section deals with the operation of modified Wallace multiplier and carry save adder respectively. In the fifth section, the obtained result for the 64 bit MAC unit is discussed and finally the conclusion is made in the sixth section



MAC Operation The Multiplier-Accumulator (MAC) operation is the key operation not only in DSP applications but also in multimedia information processing and various other applications. As mentioned above, MAC unit consist of multiplier, adder and register/accumulator. In this paper, we used 64 bit modified Wallace multiplier. The MAC inputs are obtained from the memory location and given to the multiplier block. This will be useful in 64 bit digital signal processor. The input which is being fed from the memory location is 64 bit. When the input is given to the multiplier it starts computing value for the given 64 bit input and hence the output will be 128 bits. The multiplier output is given as the input to carry save adder which performs addition. The function of the MAC unit is given by the following equation [4]: $F = IP_j Q_j$ (1) The output of carry save adder is 129 bit i.e. one bit is for the carry (128bits+ 1 bit). Then, the output is given to the accumulator register.

The accumulator register used in this design is Parallel In Parallel Out (PIPO). Since the bits are huge and also carry save adder produces all the output values in parallel, PIPO register is used where the input bits are taken in parallel and output is taken in parallel. The output of the accumulator register is taken out or fed back as one of the input to the carry save adder. The figure 1 shows the basic architecture of MAC unit.

3. IMPLEMENTATION:

Modified Wallace Multiplier A modified Wallace multiplier is an efficient hardware implementation of digital circuit multiplying two integers. Generally in conventional Wallace multipliers many full adders and half adders are used in their reduction phase. Half adders do not reduce the number of partial product bits. Therefore, minimizing the number of half adders used in a multiplier reduction will reduce the complexity [2]. Hence, a modification to the Wallace reduction is done in which the delay is the same as for the conventional Wallace reduction. The modified reduction method greatly reduces the number of half adders with a very slight increase in the number of full adders [2].

Reduced complexity Wallace multiplier reduction consists of three stages [2]. First stage the $N \times N$ product matrix is formed and before the passing on to thesecond phase the product matrix is rearranged to take the shape of inverted pyramid. During the second phase the rearranged product matrix is grouped into non-overlapping group of three as shown in the figure 2, single bit and two bits in the group will be passed on to the next stage and three bits are given to a full adder.

The number of rows in the in each stage of the reduction phase is calculated by the formula $r_{j+1} = 2[r_i/3] + r_{i \bmod 3}$ If $r_i \bmod 3 = 0$, then $r_{j+1} = 2r_i/3$ If the value calculated from the above equation for number of rows in each stage in the second phase and the number of row that are formed in each stage of the second phase does not match, only then the half adder will be used. The final product of the second stage will be in the height of two bits and passed on to the third stage. During the third stage the output of the second stage is given to the carry propagation adder to generate the final output. Thus 64 bit modified Wallace multiplier is constructed and the total number of stages in the second phase is 10. As per the equation the number of row in each of the 10

stages was calculated and the use of half adders was restricted only to the 10th stage. The total number of half adders used in the second phase is 8 and the total number of full adders that was used during the second phase is slightly increased that in the conventional Wallace multiplier. Since the 64 bit modified Wallace multiplier is difficult to represent, a typical 10-bit by 10-bit reduction shown in figure 2 for understanding. The modified Wallace tree shows better performance when carry save adder is used in final stage instead of ripple carry adder. The carry save adder which is used is considered to be the critical part in the multiplier because it is responsible for the largest amount of computation. ADDERS Addition is the most common and often used arithmetic operation on microprocessor and digital signal processor, especially digital computers. Also it serves as building block for synthesize all arithmetic operation. Therefore, regarding the efficient implementation of an arithmetic unit, the binary adder structure becomes a very critical hardware unit. In any book on computer arithmetic, someone looks that there exists a large number of different circuit arithmetic with different performance characteristics and widely used in practice. Although many researches dealing with binary adder structure have been done, the studies based on the performance analysis are only few. In this project, qualitative evaluation of the classified binary adder structures are given.

Among a huge number of the adders, verilog code for ripple carry, carry select and carry look ahead to emphasize the common performance properties belong to their classes. In the following section, we gave brief description of the studied adders architecture. With respect to the asymptote delay time and area complexity, the binary adder architecture can be characterized into four primary classes and very complex for the high bit lengths of the operands. The first class consists of the very slow ripple carry adder with the smallest area. In the second class carry select and carry skip adders with multiple levels have small area requirements and short end computation times. From the third class carry look ahead adder and the fourth class, the parallel prefix adder represent fastest addition schemes with the largest area complexities. The final step in completing the multiplication procedure is to add the final terms in the final adder. This is normally called "Vector-merging" adder. The choice of the final adder depends on the structure of the accumulation array. Following is a list of fast adders which are normally used. 1. Carry look-ahead adder 2. Simple carry skip adder 3. Multilevel carry skip adder 4. Carry-select adder 5.

Conditional sum adder 6. Hybrid adder The following subsections discuss these adders briefly. Carry look-ahead adder (CLA) The concept behind the CLA is to get rid of the rippling carry present in a conventional adder design. The rippling of carry produces unnecessary delay in the circuit. For a conventional adder the expressions for sum and carry signal can be written as follows

$$S = A \oplus B \oplus C$$

$$C_0 = AB + BC + AC$$

It is useful from an implementation perspective to define S and Co as functions of some intermediate signals G (generate), D (delete) and P (propagate). G = 1 means that a carry bit will be generated, P = 1 means that an incoming carry will be propagated to C0. These signals are computed as

$$P = A \oplus B$$

$$G = AB$$

We can write S and C0 in terms of G and P.

$$C_0(G, P) = G + PC$$

$$S(G, P) = P \oplus C$$

For an N-bit adder, the following relation holds for the carry signal.

$$C_{0,k} = G_k + P_k C_{0,k-1}$$

In a fully expanded form [1], we have

$$C_{0,k} = G_k + P_k(G_{k-1} + P_{k-1}(\dots + P_1(G_0 + P_0 C_{i,0})))$$

For k=4, the logic expressions are as

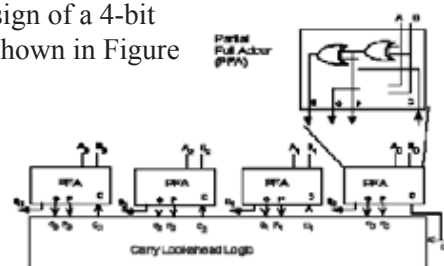
$$C_4 = G_4 + G_3 P_3 + G_2 P_2 P_3 + G_1 P_1 P_2 P_3 + C_0 P_0 P_1 P_2 P_3 = G_4 + P_3 C_3$$

$$C_3 = G_3 + G_2 P_2 + G_1 P_1 P_2 + C_0 P_0 P_1 P_2 = G_3 + P_2 C_2$$

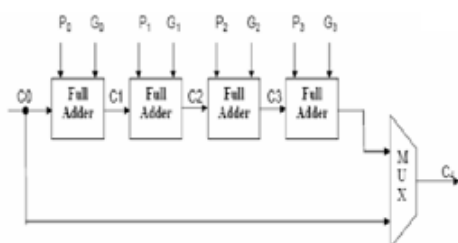
$$C_2 = G_2 + G_1 P_1 + C_0 P_0 P_1 = G_2 + P_1 C_1$$

$$C_1 = G_1 + C_0 P_0$$

This expanded relationship is used to implement N-bit adder. The carry and sum outputs are independent of the previous bits. The ripple effect has been eliminated. The logic design of a 4-bit CLA is shown in Figure



Full carry CLA is impractical for wide words. Since wide gates and large stacks display poor performance, the CLA computation has to be limited to up to 2 or 4 bits in practice. For example, the equation for C₃₁ consists of 32 product terms, the largest of which contains 32 literals. Thus the required AND and OR functions must be realized by tree networks, leading to increased latency and cost. Two schemes for managing this complexity are 1) High-radix addition 2) Multilevel look ahead. Multilevel look ahead is the most widely used technique for large CLAs.



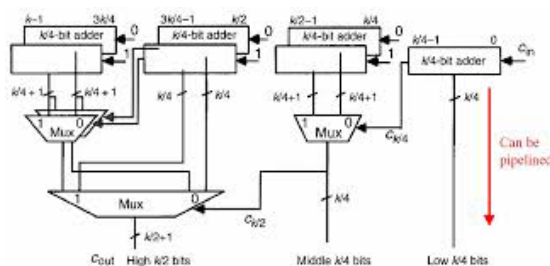
To compute the delay of an N-bit adder, we assume that there are N/M equal length bypass stages each containing M bits. An approximate expression can be derived below.

$$T_p = t_{setup} + [N/M - 1]t_{bypass} + [M - 1]t_{carry} + t_{sum}$$

t_{setup} : time to create the generate and propagate signals
 t_{carry} : propagation delay through a single bit
 t_{bypass} : propagation delay through the bypass multiplexer
 t_{sum} : time to generate the final sum

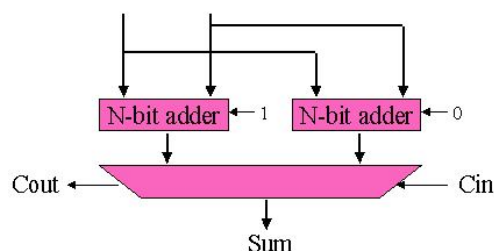
Multilevel carry skip adder :

The idea of a simple carry skip adder can be extended for N-bit adder. We assume that the total adder is divided in (N/M) equal length bypass stages, each of which contains M bits. The bits are divided in blocks of 4-bits. The diagram of 12-bit adder is shown in Figure.



4. DISCUSSION:

Carry select adder Carry select adder is based on anticipation of output carry for two possible values of input carry. Once the real value of the incoming carry is known, the correct result is easily selected with a simple multiplexer stage. Carry select adder can be implemented in two different ways 1) Linear carry select adder 2) Square-root carry select adder. Consider the block of adders, which is adding bits k to k+3. Instead of waiting on the arrival of the output carry of bit k-1, both the 0 and 1 possibility is analyzed. A multiplexer can be used to select either of the results when C₀, k-1 settles. The hardware overhead of the carry select adder is an additional carry path and a multiplexer. A basic structure of linear carry select adder is shown in Figure.



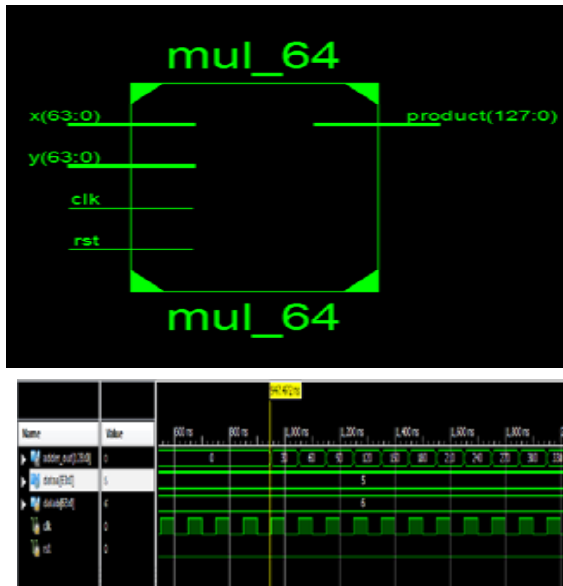
5. Concluding Remarks :

Hence a design of high performance 64 bit Multiplier-and- Accumulator (MAC) is implemented in this paper. The total MAC unit operates at a frequency of 217 MHz. The total power dissipated by 64 bit MAC unit is 177.732 mW. The total area occupied by it is 542177 11m². Since the delay of 64 bit is less, this design can be used in the system which requires high performance in processors involving large number of bits of the operation. The MAC unit is designed using Verilog-HDL and synthesized in Cadence 180nm RTL Compiler.

6. Experimental Results:

The design is developed using Verilog-HDL and synthesized in Encounter RTL compiler using typical libraries of TSMC 180nm technology. As a previous work, 8 bit MAC unit is designed using different multipliers and adders. The multipliers used for comparative study are: (i) Modified Booth Algorithm (ii) Dadda Multiplier (iii) Wallace multiplier. The different adders used in the study are: (i) Carry Look Ahead (ii) Carry Select Adder (iii) Carry Save adder.

The area, delay and power dissipation comparative table are shown in the table 1 and table 2 respectively. The graphs are plotted against the different type of 8 bit MAC unit



7.ACKNOWLEDGMENTS:

I am g.pusha and would like to thank the publishers, researchers for making their resources material available. I am greatly thankful to Assistant Prof. Miss.K.Anuradha for their guidance. We also thank the college authorities, PG coordinator and Principal for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

8.REFERENCES:

[1]. Young-Ho Seo and Dong-Wook Kim, "New VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm," IEEE Transactions on very large scale integration (vlsi) systems, vol. 18, no. 2, february 20 10

[2]. Ron S. Waters and Earl E. Swartzlander, Jr., "A Reduced Complexity Wallace Multiplier Reduction," IEEE Transactions On Computers, vol. 59, no. 8, Aug 20 10

[3]. C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. ElectronComput., vol. EC-13, no. I, pp. 14-17, Feb. 1964

[4]. Shanthala S, Cyril Prasanna Raj, Dr.S.Y.Kulkarni, "Design and VLST Implementation of Pipelined Multiply Accumulate Unit," IEEE International Conference on Emerging Trends in Engineering and Technology, ICET-ET-09 .

[5]. B.Ramkumar, Harish M Kittur and P.Mahesh Kannan, "ASIC Implementation of Modified Faster Carry Save Adder ", European Journal of Scientific Research, Vol. 42, Issue 1, 2010.

[6]. R.UMA, Vidya Vijayan, M. Mohanapriya and Sharon Paul, "Area, Delay and Power Comparison of Adder Topologies", International Journal of VLSI design & Communication Systems (VLSICSj Vo1.3, No.1, February 2012 .

[7]. V. G. Oklobdzija, "High-Speed VLSI Arithmetic Units: Adders and Multipliers", in "Design of HighPerformance Microprocessor Circuits", Book edited by A.Chandrakasan,IEEE Press,2000 .

[8]. Dadda, "Some Schemes for Parallel Multipliers," Alta Frequenza, vol. 34, pp. 349-356, 1965 .

[9]. C.S. Wallace "A Suggestion for a fast multipliers," IEEE Trans. Electronic Computers, vol. 13, no.1,pp 14-17, Feb. 1967 .

[10]. L.Dadda, "On Parallel Digital Multiplier", Alta Frequenza, vol. 45, pp. 574-580, 1976.

Author's Details:

Ms. G.PUSHPA MTech student, in M.Tech Student, Dept of ECE in KITS for women's,kodad, T.S, India.

Ms.K.Anuradha working as a Assistant at ECE in KITS for women's,kodad, T.S, IndiaJNTUH Hyderabad. she has 3 years of UG/PG Teaching Experience.