International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

Academic Science

A Review on Different Multiplier Techniques

B.Sudharani
Research Scholar, Department of ECE
S.V.U.College of Engineering
Sri Venkateswara University
Tirupati, Andhra Pradesh, India

Dr.G.Sreenivasulu
Professor and HOD, Department of ECE
S.V.U.College of Engineering
Sri Venkateswara University
Tirupati, Andhra Pradesh, India

ABSTRACT: In the present generation, VLSI systems and their design became so much important in the Electronic Engineering. As the VLSI design is the basis for electronic components, one should optimize the constraints. For good performance systems, such as digital signal processors and microprocessors, Multiplier is the important hardware block. In designing VLSI systems, the key criteria of interest are high speed, low power, less area, minimal cost. Several researchers have attempted and are attempting to design multipliers which give either of these, design targets-high speed, low power consumption, regularity of layout and hence less area as well as combination of these in one multiplier, thus making them suitable for various compact VLSI implementations. Improving speed results always in larger areas. So, this paper provides the methods that optimize the area and increases the multiplication process.

Keywords:Multiplier, Array Multiplier, Booth Multiplier, Dadda Multiplier, Wallace Tree Multiplier.

I. INTRODUCTION

In the history of digital circuits the major areas of concern in the design of digital circuits were the parameters like high speed, small area and low cost. The recent advancements in Very Large Scale Integration (VLSI) technology have resulted in high performance, battery operated, system-on-chips (SOC) in the fields of communication and computing and this has shifted the focus of the designers from traditional constraints such as area, performance, cost and reliability to power consumption. Low power VLSI circuits have become significant in energy efficient electronic designs for the high performance and portable devices.

The evolution of personal computing devices and wireless communication systems have made power dissipation is another critical design parameter. In battery operated system, the amount of energy stored within the battery is very limited. Therefore, power dissipation is important for portable systems as it defines the average lifetime of the battery. Without low power design techniques, such applications generally suffer from very short battery life. Also, packaging and cooling of those devices would be very difficult and this will lead to unavoidable increase in the cost of the product. The limited power capacity systems had given rise to more power aware designs by designers. With the advance of VLSI technology, multimedia processing and digital communication have now realized to either speed up the operation or reduce the power consumption.

II. ROLE OF MULTIPLIERS

Multiplication is a fundamental operation in modern electronics. Multiplication based functions such as Multiply and Accumulate (MAC) and inner product are some frequently employed operations in several Digital Signal Processing (DSP) applications. The ALU of microprocessors also makes use of multiplication in most of its common operations. The rapid growth in computer and signal processing applications has resulted in an increasing demand for high speed processing. Many real time signal and image processing applications depend on higher throughput arithmetic operations to achieve better performance. One of the key arithmetic operations in such applications is multiplication and the optimum design for a multiplier circuit has been a subject of interest over decades.

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

Digital multipliers are the core components of high performance systems such as Finite Impulse Response (FIR) filters, microprocessors and digital signal processors. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system and the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue.

Multiplier is the main key structure for designing an energy efficient processor and a multiplier design decides the efficiency of a digital signal processor. It forms to be one of the basic building blocks of Digital Signal Processing applications. In many DSP algorithms, the multiplier lies in the critical delay path and ultimately decides the performance of the algorithm. The speed of the multiplier is of greater importance in DSP as well as in a general processor. Thus, multiplier performance is the most critical factor in every computational system. Due to the importance of digital multipliers in DSP, it has always been an active area of research. So reducing the power of multipliers will reduce the power of digital signal processors.

III. FUNDAMENTALS OF MULTIPLIERS

A simple multiplier contains three parts as shown in figure1:

i) Partial Product Generation

ii) Partial Product Accumulation and

iii) Final Additions.

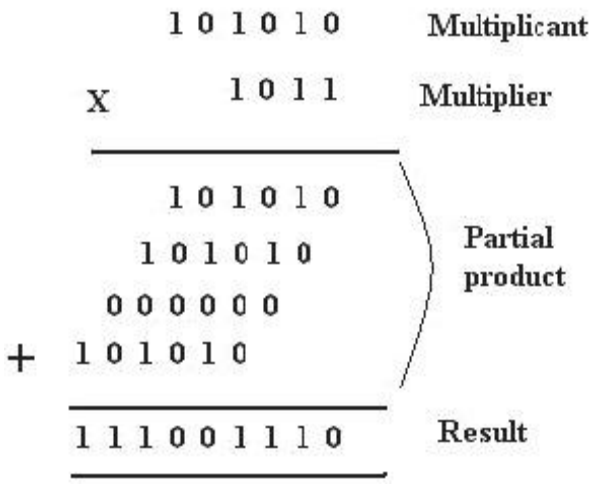

Fig 1: Structure of a Multiplier

A multiplier essentially consists of two operands, a multiplicand and a multiplier which produces a product. In the first stage, the multiplier and the multiplicand are multiplied bit by bit to generate the partial product. The second stage is the most important as it is the most complicated and determines the speed of the overall multiplier to add these partial products to generate the product.

The simplest way to perform a multiplication is to use a single half adder. For inputs that are M and N bits wide the multiplication tasks M cycles using an N-bit adder. This shift-and-add algorithm for multiplication adds together M partial products. Each partial product is generated by multiplying the multiplicand with a bit of the multiplier and by shifting the result in the basis of the multiplier bit's position. Similar to the familiar long hand decimal multiplication, binary multiplication involves the addition of shifted versions of the multiplicand based on the value and position of each of the multiplier bits. As a matter of fact, it is much simpler to perform binary multiplication than decimal multiplication. The value of each digit of a binary number can only be 0 or 1.Thus, depending on the value of the multiplier bit, the partial products can only be a copy of the multiplicand or 0. In digital logic this is simply an AND function.

Multipliers are in fact complex adder arrays. This is an operation common to a large number of applications, and the complexity of this function has lead to a large amount of research directed at speeding up its execution. Multipliers can be implemented using different algorithms. Depending on the algorithm used, the performance characteristics of the multipliers vary. In the implementation of digital multipliers binary adders are an essential component. With the emergence of power as a design consideration, speed is not the only criterion by which various implementations are judged. Designing multipliers with low power energy efficient adders minimize the power consumption and delay of the multipliers.

IV. SIGNIFICANCE OF LOW POWER MULTIPLIERS

As the scale of integration keeps growing, more and more sophisticated signal processing systems are now being implemented on a VLSI chip. The majority of the signal processing applications not only demand great computation capacity but also consume considerable amount of energy. Hence, power consumption has become a critical aspect in today's VLSI system design. The need for low-power circuits arises from two main factors:

- First, due to high operating frequency and processing capacity per chip, delivering of large currents, heat emitted due to large power consumption and the need for proper cooling techniques.
- Second, due to the limited battery life in portable electronic devices.

Low power design directly leads to prolonged operation time in these portable devices. Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore, design of low power multipliers play an important role in low power VLSI system design.

V. NEED FOR RESEARCH IN LOW POWER MULTIPLIERS

Multiplier features as a crucial element of the digital signal processor and in several other applications. The efficiency of the digital signal processor could be increased only based on the performance of the multiplier. With innovations in technology several researchers have attempted and are attempting to design multipliers which provide either of the following design targets; high speed, low power consumption, regularity of layout, less area and even mixture of these in a single multiplier. Hence making them ideal for various high speed, low power and compact VLSI implementation. A multiplier is among the key hardware blocks in many digital signal processing (DSP) systems. Typical DSP applications where a multiplier plays an important role include digital filtering, digital communications and spectral analysis.

Many current DSP applications are targeted at portable battery operated systems so the power dissipation becomes one of the primary design constraints. Since multipliers are rather complex circuits and must typically operate at a high system clock rate, so reducing the delay of a multiplier is an essential part of satisfying the overall design. Moreover Application Specific Integrated Circuits (ASICs) rely on efficient implementation of the arithmetic circuits required in the execution of the algorithms. Also, the upsurge in the number of transistor effects in the increase in the complexity of the arithmetic circuits thus leading to more power consumption.

Consequently, the design of low power multipliers is a necessity for the design and implementation of efficient power aware devices. This serves as a motivation in the design of low power VLSI circuits. But, reducing the power consumption and enhancing the circuit design pose challenges in the multiplication and accumulation. Hence, the need for research on dynamic power reduction and static power reduction in the design of circuits become evident. Researchers have recognized the importance of designing low power multipliers. Some research has resulted in low power circuit designs formultipliers. This research has focused on designing low power multipliers using four such circuit designs.

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

Academic Science

VI. METHODS

Multiplication is a lot common operation than addition, and is needed for microprocessors, digital signal processors and graphics engines. The most basic form of multiplication consists of forming the product of two unsigned (+) numbers. Many different kinds of multipliers have been proposed with very different hardware requirements, throughput and power dissipation. These include: serial multipliers, sequential multipliers, array multipliers and tree multipliers.

Serial multipliers and sequential multipliers are rarely used in today's high-performance CMOS circuits because of their poor throughput, although they are quite power-efficient. Array multipliers and tree multipliers are two of the most popular kinds of multiplier. The basic principles of array and tree multipliers are:

- Generate partial products;
- Add all the partial products together through several rows of carry-save adder (CSAs) using, for example, 3-2 (full) adders or 4-2 adders, finally obtaining one partial sum and one partial carry;
- Send the partial sum and partial carry to a multi-bit carry-propagate adder to get the final result.

Array multipliers and tree multipliers are fast but expensive in terms of hardware and power consumption. Iterative structures allow a trade-off between performance and hardware requirement. Pipelining is usually used in iterative systems to improve their performance.

There are a number of techniques that can be used to perform multiplication. In general, the choice is based up on factors such as latency, throughput, area, and design complexity. An obvious approach is to use an M+1 – bit carry propagate adder (CPA) to add the first two partial products, then another CPA to add the third partial product to the running sum, and so forth. Such an approach requires N-1 CPA's and is slow, even if a fast CPA is employed. More efficient parallel approaches use some sort of array or tree of full adders to sum the partial products.

In early 1950's, multiplier efficiency was considerably improved with the introduction of booth multiplier and the growth of faster adders and memory components. Booth's method and the modified booth's method don't require a correction of the product when either (or both) of the operands is negative for two's complement numbers. During the 1950's, adders designs moved away from the slow sequential formation of carried executed by ripple carry adders carry look ahead, carry select, and conditional some adders gave speedy sums through the faster simultaneous or parallel generation of carriers.

In the 1960's two classes of parallel multipliers were described. The first class of parallel multipliers uses a rectangular array of similar combinational cells to make and sum the partial product bits. Multipliers of this kind are called Array multipliers. They have a delay that's generally proportional to the word length of the multiplier input. Because of the regularity of their structures, array multipliers are carrying to layout and have now been applied frequently. The second class of parallel multipliers reduces a matrix of partial product bits to two words through the strategic application of counters or compressors. Both of these words are then summed utilizing a fast carry-propagate adder to generate the product. This class of parallel multiplier is called as Column compression multiplier. Because the delay is proportional to the logarithm of the multiplier, word length they are also the fastest multipliers.

In array multiplier, the two simple functions of partial product generation and summation are combined. For unsigned N by N multiplication, N2+N-1 cells, where N2 contain an AND gate for partial product generation and a full adder for summing and N-1 cells comprising a full adder, are connected to produce a multiplier. The array generates N lower product bits directly and uses a Carry-propagate adder, in cases like this a ripple carry adder, to form the upper N bits of the product.

In order to design an array multiplier for two's complement operands, Booth algorithm could be employed. The implementation of a booth's algorithm array multiplier computes the partial products by analyzing two multiplicand bits at a time. Except for enabling usage of two's complement operands, this booth's algorithm array multiplier offers no performance or area gain when compared with the basic array multiplier. Better delays, however may be achieved by implementing a higher radix modified booth algorithm.

Yet another method for building an array multiplier that handles two's complement operands was presented by Baugh and Wooley. This technique increases the maximum column height by two. This can lead to an additional stage of partial product reduction, thus increasing overall delays.

A modified form of the Baugh and Wooley technique is more frequently used since it does not increase the maximum column height. Column compression multiplier continued to be studied because of their high speed performance. With total delays that are proportional to the logarithm of the operand word length, column compression multipliers are quicker than array multipliers whose delay grows linearly with operand word length.

According to Thomas Ko Callaway and Earl E. Swartzlander. Jr [1], column compression multipliers are more power efficient than array multipliers. In 1964, Wallace presented a scheme for fast multiplication based on summing the partial product bits on parallel using a tree of carry save adders which became generally known as the Wallace tree .

Dadda later enhanced Wallace's method by defining a counter placement strategy that needed less counters in the partial product reduction stage at the cost of a larger carry-propagate adder. For both methods, the total delay is proportional to the logarithm of the operand word-length. Other partial product reduction methods have now been proposed since the work of Wallace and Dadda. The reduced area and the Windsor methods are based on strategic utilization of (3, 2) and (2, 2) counters to improve area and layout, while maintaining the fast speed of the Wallace and Dadda designs.

Londono.S.M [11] proposed a new adaptive energy-aware system design methodology for a 32-bit multiplier with16-bit and 32-bit precision; and variable performance. The configurable-reuse of point solution method is described and evaluated. Results show up to 80% energy savings for 16-bit and up to 60% for 32-bit operations over a static 32-bit multiplier and up to 22 % in average compare with a twin precision multiplier (CMOS 65-nm technology).

Ruan.A.W [10] traditionally, each time a new finite impulse response (FIR) filter is required to design, a new algorithm have to be developed specially for the FIR filter and ALU based FIR filter hardware architecture must be designed to meet the design specifications. Rather than multiplier-accumulator based architecture used in conventional FIR, the proposed ALU architecture implements FIR (with 64-taps) functions by using accumulators and shift-registers controlled by the instructions of ROM. TDMA technique is employed to reduce chip size.

Ramanthan P [8] discusses the method, to investigate the area and power-delay performances of 8 bit and 16 bit multiplier combining both Wallace tree and decomposition algorithm improves speed and reduces power consumption by reducing the spurious transitions on internal nodes and analyzed area and power-delay performances for adder structures using different logic families CMOS, HYBRID AND CPL.

Hussin.R [7] presents the design of an efficient multiplication architecture based on radix 4 booth multiplier (with 2 enhancements). The first is to modify the Wen-Chang's modified booth encoder (MBE) with the simplified sign extension (SSE) method. The 2nd part is to improve the delay of the carry signal in the 4:2 compressor circuit. The Gajski rule 32 has been adopted in order to estimate the delay and size of the circuit. The total transistor count for this new multiplier is being a slightly bigger. In this design the propagation delay is reduced by 2 to7% from existing designs.

Cardarilli. G.C [5], proposed the use of signed digit (SD) arithmetic to better exploit the architectural characteristics of FPGAs. The implementation of Radix-4 SD adders, multipliers and FIR filters has been carried out to demonstrate that the use of this number system representation optimally fits the 6-input LUT Logic Elements (LEs) of the newest FPGAs architectures. From the results obtained based on 4/6 -input LUT based FPGAs; Radix-4 SD arithmetic is very efficient.

Athow. J.L [6] presents a partition algorithm for large integer multipliers with speed as optimization criteria and uses built-in high-speed arithmetic blocks available in the current generation of Xilinx FPGA chip. The proposed technique reduces the delay more than 30% compared to both Xilinx Coregen and Xilinx Synthesis models.

Ngo. H.T [4], proposed a neighborhood dependent approach (NDA) for the design of a high performance and low power radix-4 booth multiplier for kernel-based operations such as 2D convolution in video processing applications to reduce dynamic power consumption by analyzing the bit patterns in the input data to reduce switching activities. Special values of the pixels in the video streams such as zero, repeated values or repeated bit combinations are detected and data paths in the architecture design are disabled appropriately to eliminate unnecessary switching in arithmetic units and data buses. Input pixels in the video stream are partitioned into halves to increase the possibility of detecting special values.

It is observed that the proposed scheme helps to reduce operations and switching activities in the 2D convolution operations up to 46% of the switching activity rate which results in significant power reduction with low hardware overhead.

VII. CONCLUSIONS

Table 1 reveals the comparison of speed, complexity, layout, area, power and cost of varied types of multiplier architectures.

Table 1.COMPARISON OF MULTIPLIER ARCHITECTURES

Type	Speed	Complexity	Layout	Area	Power	Cost
Array	Low	Simple	Regular	Smallest	High	Low
Carry Save	Medium	Simple	Regular	Smaller	High	Average
Booths	High	Medium	Irregular	Medium	Medium	Medium
Wallace Tree	Higher	Medium	Irregular	Medium	Medium	Average
Dadda	Higher	Medium	Irregular	Medium	High	Average

From the literature review carried out on various multipliers, it is found that the multiplier architecture basically consists of adders. Hence designing a high speed adder or high speed full adder enhances the performances of multiplier architecture. The partial products computed by the adders in a multiplier and hence placement and routing of adders shall also influence the area, power and delay variables of multipliers. From the review it is found that Wallace tree multiplier is recommended and is adopted for numerous high speed applications.

Acknowledgment

I want to express my sincere thanks to my Family Members & Colleagues for their constant help, valuable guidance and useful suggestions, which helped me in the successful completion of the work.

Academic Science

REFERENCES

[1] Thomas K. Callaway and Earl E. Swartzlander, Jr. (1993), "Optimizing Multipliers for WSI", Int.Conf., on Wafer Scale Integration, pp. 85 - 94, 1993

[2] C. S. Wallace (1964), "A Suggestion for a Fast Multiplier", IEEE Trans., Electronic Computers, vol. EC - 13, pp. 14 - 17, 1964.

[3] Luigi Dadda (1976), "On Parallel Digital Multipliers", Alta Frequenza, vol. 45, pp.574 – 580.

[4] Ngo. H.T, Asari.V.K(2007), "Design of a radix-4 Booth multiplier withNeighborhood Dependent Approach for video processing applications", MWSCAS,50th Midwest Symp., Circuits and Systems, pp. 53 – 56, Aug 2007.

[5] Cardarilli. G.C, Pontarelli. S, Re. M, Salsano.A(2008), "On the use of signed digit arithmetic for the new 6-inputs LUT based FPGAs", 15th IEEE Int.Conf., ICECS, pp. 602 – 605, Aug 2008.

[6] Athow.J.L, Al-Khalili.A.J(2008), "Implementation of large-integer hardware multiplier in Xilinx FPGA", 15th IEEE Int.Conf., ICECS, pp. 1300 – 1303.

[7] Hussin.R, Shakaff. A.Y.M, Idris. N, Sauli. Z, Ismail.R.C, Kamarudin. A (2008), "An efficient Modified Booth multiplier architecture", ICED, p. 1 – 4, Dec. 2008.

[8] Ramanthan.P, Vanathi. P.T, Chaubey. A, Senthilraja. N (2009), "Decomposition algorithm for power delay product optimization in Wallace multiplier", Int.Conf., Control, Automation, Communication and Energy Conservation, pp. 1 – 6, 2009.

[9] Gomes. M, Silva. V, Cercas. F, Tomlinson. M (2009), "Real-time LUT - less magnitude modulation for eak power control of single carrier RRC filtered signals", IEEE 10th Workshop on Signal Processing Advances in Wireless CommunicationsSPAWC, pp. 424 – 428, June 2009.

[10] Ruan. A.W, Liao. Y.B, Li. P, Li. J.X (2009), "An ALU-based universal architecture for FIR filters", Int.Conf., Communications, Circuits and Systems, pp. 1070 – 1073,July 2009.

[11] L. S.M, de Gyvez. J.P (2009), "An energy-aware multiplier based on a Configurable-Reuse of points design methodology", 16th IEEE Int.Conf., Electronics, Circuits, and Systems, pp. 89 – 92, Dec. 2009.