International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp 38572-38582

A Detailed Survey on Various Efficient Multipliers in Low Power VLSI Circuit

M.Srinivasan

Assistant Professor, Department of Electronics and Communication Engineering,
KGiSL Institute of Technology, Coimbatore.
kmsrinivasan@gmail.com

P.Prakasam,

Professor, Department of Electronics and Communication Engineering,
United Institute of Technology, Coimbatore.
prakasamp@gmail.com

Abstract
For energy efficient and high performance design, the low power VLSI circuit is used. Multiplier is an essential part of low power VLSI design, since the efficiency of the digital signal processor depends upon the Multiplier. Multiplier requires more hardware resources and processing time than addition and subtraction process. In multiplier circuit, most of the power is dissipated across in full adder circuits. In this paper, a detailed analysis of different types of Multiplier has been studied. The effectiveness and efficiency of these multipliers are stated in terms of performance measures such as Area, Power and Maximum delay. Based upon the application environment, an optimum value Multiplier has been chosen.

Keywords: Multiplier; VLSI design; Low power consumption; full adder.

Introduction

Now a days each circuit to face the power consumption issue for booth portable device aiming to life of the battery. The multiplication is an important fundamental of arithmetic logic operation. In DSP system multiplication process are limited and it dominates the execution time. This has begun to change and increasing i.e., power is being given comparable weight to area and speed considerations. The power consumption is reduced by decrease in feature size and increase in chip density and operating frequency. Extensive work has been carried out on low power multipliers at technology, physical, circuit and logic levels. These low-level techniques are not unique to multiplier modules and they are generally applicable to other types of modules. Moreover, power consumption is directly related to data switching patterns. However, it is difficult to consider application-specific data characteristics in low-level power optimization [1]. Digital computer arithmetic of logic design with developing appropriate algorithms by using available hardware [2-5]. In a signal processing application the high degree of parallelism and it is dominated by a few kernels such as multiplication. It is responsible for the large fraction of execution time and energy. Multiplier is a fundamental arithmetic unit and it is a basic component in a single chip digital information process. In a real time system require area, delay and power optimal architecture. So the power dissipation rises so the removal of heat becomes difficult and expensive. A multiplier uses Booth's algorithm [6] array of full adder [FAS] or Wallace tree [7]. There is wide range of multipliers Based on the way the data is processed, they are classified as serial, parallel and serial-parallel multipliers as shown in figure 1. Among these multipliers, the basic add-and-shift multiplier performs operations similar to the manual method of doing multiplication. The data are entered serially resulting in slower multiplication. In parallel multipliers, there are two main classifications. They are array and tree multipliers. C.S. Wallace proposed a tree multiplier architecture which performs high speed multiplication. But this has a high structural irregularity and is unsuitable for VLSI implementation as it demands regularity. Braun multiplier is an unsigned parallel array multiplier which requires n2 and gates and n-1 adders for the multiplication of n bit operand. They are unsuitable for large size operands..

The objective of realizing a good multiplier is to have a small size, high speed and low power consumption. To save significant power consumption of a VLSI design, the focus to be to reduce its dynamic power, the bulk of total power dissipation. The multiplier mainly consists of three points: Booth encoder a tree to compress the partial product such as Wallace tree, and final adder [8]. In first stage, Multiplication is implemented by accumulation of partial products, each of which is conceptually produced via multiplying the whole multi-digit multiplicand by a weighted digit of multiplier. To compute partial products, most of the approaches employ the Modified Booth Encoding (MBE) approach [9], for the first step because of its ability to cut the number of partial products rows in half. In next step the partial products are reduced to a row of sums and a row of caries which is called reduction stage The various types of multipliers are available in VLSI design such as basic multiplication scheme, Booth multiplier, tree and Array multiplier, Wallace Tree Multiplier, Mixed Style multiplier, Bit-Serial multiplier, modular multiplier. The multiplication is considered as a series is the multiplicand. The number of times that is added is the multiplier and the result is product. The product is generally twice the length of

operands. The multiplication is decomposed in to two parts such as generation of partial products and second one is collects and adds them based on the architecture, application and the way of producing partial products and summing in of partial products is dedicated to the generation of partial products, and the second one collects and adds them.

The multiplier are categorized the continuous development of VLSI design the chip design tools are growing [10], [11], [12]. The speed, power and area are the used measure of an algorithm. Since ultimately, speed, power and chip area are the most often used measures of the efficiency of an algorithm, and there is a strong link between the algorithms and technology used for its implementation.

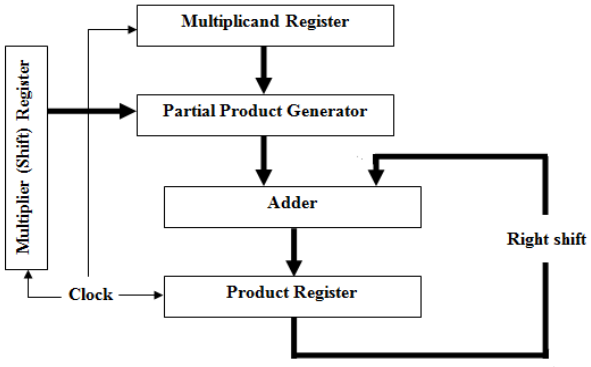

Fig.1. General Implementation of Multiplier

Detailed Problem Description

In a digital circuit the problem of estimating the average power dissipation referred to as power estimation. This is mainly different from estimating the worst case instantaneous power. It referred to as the voltage drop problem [13]-[15]. A multiplier design consists of three operational steps. The first is radix-2 Booth encoding in which a partial product is generated from the multiplicand X and the multiplier Y. The second is adder array or partial product compression to add all partial products and convert them into the form of sum and carry. The last is the final addition in which the final multiplication result is produced by adding the sum and the carry. Average power is directly related to chip heating and temperature. We already know the straight forward method of power estimation is simulation of design and monitor the power supply current. The average of the current waveform is computed and provides the average power.

The accuracy and generality are the main advantage of this technique. It is mainly used to estimate the power, regardless of technology, design style, functionally, architecture etc. Simulation result is directly related in the input signal. The complete information about the input signal is required in the form of voltage waveform. In the simulation based technique the pattern-dependence problem will occur. The power of a functional block needs to be estimated [16]. When the rest of the chip not yet been designed. Input are typical, large numbers of input pattern had to be simulated. This can become very expensive, practically impossible for large circuits. These problems can be sampling in three ways, first one is the power supply and ground voltage levels of the power supply and ground voltage levels of the chip are fixed. So that it become simpler to compare the power by estimating the current by every sub-circuit. Second one is the circuit is built of logic gates and latches. It is a well structured design style by a common clock and combinational logic blocks whose inputs are latch output latches are edge-triggered and the use of a CMOS or BiCMOS design technology. The average power dissipation of the power The Original version of Booth's multiplier (Radix – 2) had two drawbacks. The number of add / subtract operations became variable and hence became inconvenient while designing Parallel multipliers. The Algorithm becomes inefficient when there are isolated 1s.

Array Multiplier is the layout of the combinational multiplier. Multiplication is the series of repeated additions. The number to be added is the multiplied, the number of times that is added is the multiplier and the result is the product in each step of addition generates partial product. The multiplication of two binary number can be obtained with one micro-operation by using combinational circuit, the circuit forms the product but thus it making fast way of multiply two numbers, the multiplication array is formed by only delay is the time for the signals to propagate through the gates. Array multiplier gives more power consumption because of the optimum number of components required, when the area of the multiplier is increased and it requires large number of gates because of the delay for this multiplier is larger. The array multiplier is a fast multiplier, but hardware complexity is high and it is less economical [17]-[19].

The Wallace tree is a tree of carry-save adders. Carry-save adder consist of full address like the ripple adders, the carry output from each bit is brought out to form second result vector rather than wired to the next most significant bit. The carry vector is 'saved' to be combined with the operation is high. Since the circuit is irregular and also the circuit layout is not easy. [17] Another one fast process for multiplication of two numbers was developed by Wallace [20]. In this method multiplying two numbers by using three steps. In the first step the bit products are formed, the second stages is bit product matrix is reduced to a two rows matrix, when sum of the now equals the sum of bit products. The third stage is the two resulting rows are added with a fast adder to produce the final product.

Another method of improvement in multiplier is by reducing the number of partial products generated, by using booth recoding multiplier [21]. It reduce the number of partial products by the scan the three bits at a time. The three bits are such as the two bit from the present pair; and the third bit from the higher order bit of an adjacent lower order pair. After examining each triplet of bits, the triplets are converted by booth logic into a set of five control signal used by the adder. The adder cell performed by cell in the array to control the operations. Booth recoding reduces the number of partial products such as reducing the number of adders and also the delay. It required to produce the partial sums by examine three bits at a time. The high performance of booth multiplier is introduced the one drawback of this multiplier is the power consumption. Because of large number of adder cells are formed, it consumes large power [21].

Brief Overview

The power estimation is generally referred to as the problem of estimating the average power dissipation. The estimating power dissipation is strongly pattern dependent circuit simulation based [22], [23]. Once the circuit will be simulated while monitoring the supply voltage and current wave-forms. This is used to compute the average power. These are strongly patterned depended, these circuits are too slow to be used in a large circuit, for which the high power dissipation is one of the problem. Another simulation technique are also proposed o improve the computational efficiency. These techniques are proposed using various kinds of timing, suiton level and logic simulation [24], [25]. This Booth multiplier technique is to increase speed by reducing the number of partial products by half. Since an 8-bit booth multiplier is used in this project, so there are only four partial products that need to be added instead of eight partial products generated using conventional multiplier. The power supply and ground voltage are fixed, and the supply current waveforms estimated by using these techniques. While it indeed more efficient compared to the traditional circuit simulation, it is strongly pattern dependent because of cost of some loss in accurate. Based on the latency, throughput area, and design complexity, the number of techniques are introduced to perform the binary multiplication. Most efficient parallel approach uses some gong of array or tree of full adders to sum partial products. The some of the standard approaches to have hardware implementation of binary multiplier are array multiplier, booth multiplier and Wallace tree multipliers. Which are suitable for VLSI implementation at CMOS level.

A. Array Multiplier

Array multiplier is an efficient layout of a combinational multiplier. Multiplication is an important fundamental function in arithmetic logic operation. An array multiplier is very regular in structure. It uses short wires that go from one full adder to adjacent full adders horizontally, vertically or diagonally [14]. An n × n array of AND gates can compute all the terms simultaneously. The terms are summed by an array of 'n [n-2]' full adders and 'n' half adders. The shifting of partial products for their proper alignment is performed by simple routing and does not require any logic. The multiplication dominates the execution time of most DSP algorithm. Multiplication can be considered as a series of repeated addition. By using combinational circuit the multiplication of two binary numbers can be obtained with one-micro-operation that forms the product bit, all at once thus making it a fast way of multiplying two numbers. Since only delay is the time for the signals to propagate through the gates that forms the multiplied array. In array multiplier, we consider two binary numbers A and B of bits M and N respectively.

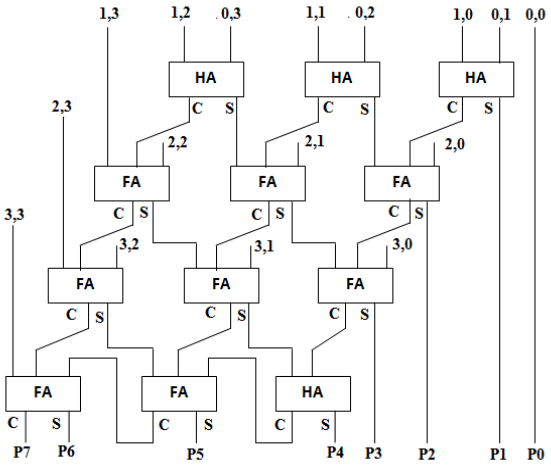

Fig.2. Array Multiplier

The sum of this bits are Mn and that are produced in parallel by a set of Mn AND gates. The nxx multiplier consist of (n-2) full adders, n-half-adders and n2 AND gates. The worst case delay of array multiplier is (2n+1) td. The delay of the array multiplier is larger and it gives more power consumption as well as optimum number of components required. In array multiplier forms larger number of gates because of the area of the multiplier is increased. So the array multiplier is less economical [17]-[19]. One of the advantages of array multiplier is its fast multiplier and also the demerits of array multiplier are the hardware complexity is high.

B. Wallace Tree Multiplier

The multiplication of two numbers was developed by Wallace [20]. Using this method the multiplication of two numbers is considered in three steps. The amount of hardware required to perform this style of multiplication is large but the delay is near optimal. The delay is proportional to log (N) for column compression multipliers where N is the word length. This architecture is used where speed is the main concern not the layout regularity. This class of multipliers is based on reduction tree in which different schemes of compression of partial product bits can be implemented. The first stage is bit products are formed and the second stage is the bit product matrix is reduced to a two row matrix, where sum of the row equals the sum of bit product and the third stage is the two resulting rows are added with a fast adder to produce final product. The three input Wallace tree circuits is a circuit of three bit signals are passed to a one bit full adder ("3W') and the output signal is supplied to the next stage full adder of the same bit, and the carry output signal is passed to the next stage full adder of the same no. of bit, and the carry output signal is supplied to the next stage full adder of the same bit, and the carry output signal is supplied to the next stage of the full adder located at a one bit higher position. The Wallace tree multiplier is considerably faster than a simple array multiplier because its height is logarithmic in word size, not linear. However, in addition to the large number of adders required, the Wallace tree's wiring is much less regular and more complicated. As a result, Wallace trees are often avoided

by designers, while design complexity is a concern to them. Wallace tree styles use a log-depth tree network for reduction Faster, but irregular, they trade ease of layout for speed. Wallace tree styles are generally avoided for low power applications, since excess of wiring is likely to consume extra power. While subsequently faster than Carry-save structure for large bit multipliers, the Wallace tree multiplier has the disadvantage of being very irregular, which complicates the task of coming with an efficient layout.

The Wallace tree multiplier is a high speed multiplier. The summing of the partial product bits in parallel using a tree of carry-save adders became generally known as the "Wallace Tree". Generally Wallace tree is a tree of carry-save adders arranged as shown in figure 3. A carry save adder consist of full adder like ripple adder.The carry-save adder output from each bit is brought out to form second result vector rather than the wired to the next most significant bit.In the Wallace tree multiple the speed of the operation is high, since the circuit is quite irregular, and also in Wallace tree method, the circuit layout is not easy.

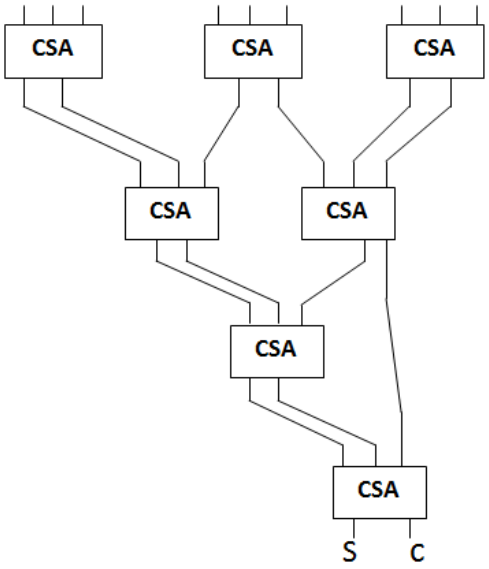

Fig.3. Wallace Tree Multiplier

C. Booth Multiplier

Another methods of improving the multiplier by reducing the partial products by using Booth Multiplier. This type of multiplier operates much faster than an array multiplier for longer operands because its computation time is proportional to the logarithm of the word length of operands. Booth multiplication is a technique that allows faster multiplication by grouping the multiplier bits. The grouping of multiplier bits and Radix-2 Booth encoding reduce the number of partial products to half. Booth recording multiplier is one of the multiplier; Booth multiplication [16] is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is the standard technique used in chip design, and provides significant improvements over the "long multiplication" technique. it scans the three bits at a time to reduce the number of partial products [21]. The architecture of a radix 2^n multiplier the multiplication of two numbers with four digits each these numbers are denoted as V and U while the digit size was chosen as four bits. The reason for this will become apparent in the following sections. Each circle in the figure 4 corresponds to a radix cell which is the heart of the design. Every radix cell has four digit inputs and two digit outputs. The input digits are also fed through the corresponding cells. The dots in the figure represent latches for pipelining. Every dot consists of four latches. The ellipses represent adders which are included to calculate the higher order bits. They do not fit the regularity of the design as they are used to "terminate" the design at the boundary. The outputs are again in terms of four bit digits and are shown by W's. The 1's denote the clock period at which the data appear Booth multipliers save costs (time and area) for adding partial products. With the higher radix the number of additions is reduced and the redundant Booth code reduces costs for generating partial products in a higher radix system.

Low power consumption is there in case of radix 4 booth multiplier because it is a high speed parallel multiplier. Though Wallace Tree multipliers were faster than the traditional Carry Save Method, it also was very irregular and hence was complicated while drawing the Layouts. If we compare the above values among each other we can observe that the Array Multiplier is the worst case multiplier consuming highest amount of power and then comes the Radix – 2 booth multiplier which consumes lesser power than array multiplier. The Wallace Tree multiplier and Booth Multiplier Radix-4 have nearly same amount of delay while Radix-4 Booth consuming lesser power than the other. Hence we reach to a conclusion that Booth Radix-4 Multiplier is best for situations requiring Low power Applications. Slowly when multiplier bits gets beyond 32-bits large numbers of logic gates are required and hence also more interconnecting wires which makes chip design large and slows down operating speed Booth multiplier can be used in different modes such as radix-2, radix-4, radix-8 etc. But we decided to use Radix-4 Booth's Algorithm because of number of Partial products is reduced to n/2.

The main reason for the power consumption is large number of adder cells required that consumes large power [21]. A radix-4/-8 multiplier is implemented using modified booth multiplier encoder that demand high speed and low energy operation. Depending on the input pattern, the multiplier operates in the radix-8 mode in 56% of the input cases for low power, but reverts to the radix-4 mode in 44% of the slower input cases for high speed. The performance of the radix-8 multiplier is bottlenecked due to the occurrence of the 3B term in computing the partial products. So for computing the partial product in this case we select the radix-4 mode [26].These 3 bits are such that the two bit from the present pair and a third bit from the high order bit of an adjacent lower order pair.Examining each triplet of bits, then the triplets are converted by Booth logic into a set of fuse control signals used by the adder cells in the array to control the operation performed by the adder cells.The booth multiplication can be proved that the addition/subtraction operation can be skipped, if the successive bits in the Multiplicand are same. The addition and subtraction operation can be skipped, in the condition of 3 consecutive bits are same.

In some cases the delay associated with Booth Multiplication are smaller than compared to the Array Multiplier, the input data is dependent on the performance of Booth Multiplier delay. In the worst case the delay with booth multiplier is with array multiplier [27]. The method of Booth Recording reduces the number of adder and hence the delay required to produce the partial sum by examining three bits at a time. The main drawback of the multiplier is power consumption.

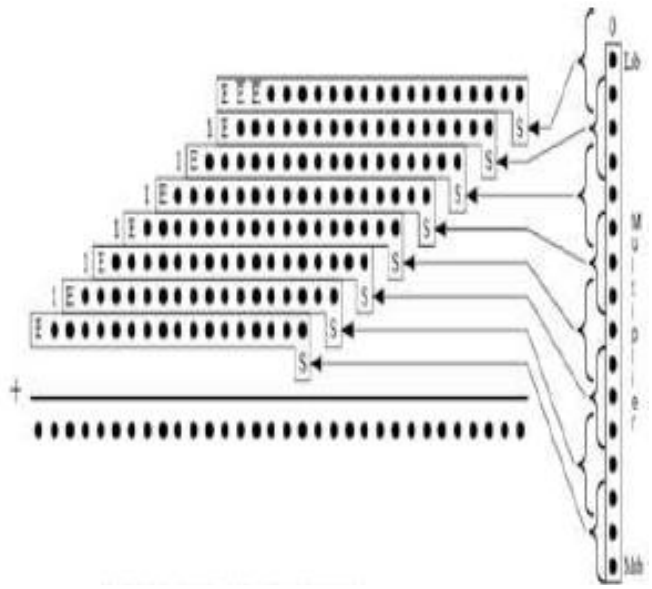

Fig. 4. Modified Booth Encoder

Grouping considered each 3 bits starts from LSB and first considered only two bits, next considered 3 bits but one bit will be overlapped on the previous group,

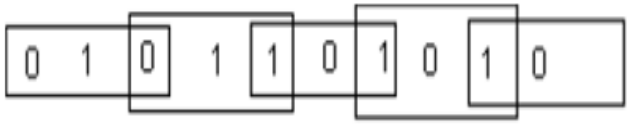

Fig.5. Grouping of bits from the multiplier term

The advantages of the modified booth multiplier is reducing the number of partial products in to half of the multiplier term size by grouping, complexity of the circuit to produce partial product is the main disadvantages of this multiplier.
In order to achieve high-speed multiplication, multiplication algorithms using parallel counters, such as the modified Booth algorithm has been proposed, and some multipliers based on the algorithms have been implemented for practical use. This type of multiplier operates much faster [39] than a tree multiplier for longer operands because its computation time is proportional to the logarithm of the word length of operands [40].

Table1. Recoding Table

Block	Re - coded digit	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

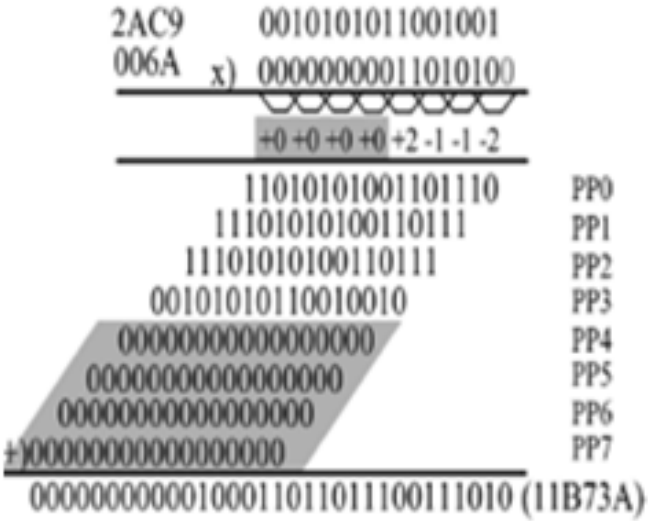

Fig. 6. Example of Modified Booth Multiplier

The figure 6 shows a computing example of Booth multiplying two numbers"2AC9" and "006A". The shadow denotes that the numbers in this part of Booth multiplication are all zero so that this part of the computations can be neglected. Saving those computations can significantly reduce the power consumption caused by the transient signals. According to the analysis of the multiplication shown in fig. 3 the SPST equipped modified-Booth encoder is proposed, which is controlled by a detection unit. The detection unit has one of the two operands as its input to decide whether the Booth encoder calculates redundant computations. The advantage of this method is making the number of partial products into half of the multiplier term size by grouping. The main disadvantage of the modified booth multiplier is its complexity of the circuit to produce partial product.
Modified Booth algorithm has been proposed for high speed multiplication.This type of multiplier operates much faster

than an array multiplier for longer operands because its computation time is proportional to the logarithm of the word length of operands. Booth multiplication is a technique that allows faster multiplication by grouping the multiplier bits. The grouping of multiplier bits and Radix-2 Booth encoding reduce the number of partial products to half. So we take every second column, and multiply by ±1, ±2, or 0, instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0.The advantage of this method is halving of the number of partial products. For Booth encoding the multiplier bits are formed in blocks of three, such that each block overlaps the previous block by one bit. Start from the LSB for grouping, and the first block only uses two bits of the multiplier.

D. Bit-Serial Multipliers

Another type of multiplier is the bit serial array multiplier to reduce the power consumption. To improve the performance at low power consumption by using pipelined method. The pipeline consists of 3 stages that is fetch, decodes and executes. In this technique it requires smaller pin count. So it enables to run at high-clock rate. In VLSI design to reduce the area, power and improve the performance, so the VLSI is needed only in systolic arrays. The diagram represents the systolic design in bit serial multiplication process.

In digital multimedia, bit rate often refers to the number of bits used per unit of playback time to represent a continuous medium such as audio or video after source coding (data compression).[41] The encoding bit rate of a multimedia file is the size of a multimedia file in bytes divided by the playback time of the recording (in seconds), multiplied by eight. For real-time streaming multimedia, the encoding bit rate is the good put that is required to avoid interrupt:

Encoding bit rate = required good put

The term average bit rate is used in case of variable bit rate multimedia source coding schemes. In this context, the peak bit rate is the maximum number of bits required for any short-term block of compressed data. A theoretical lower bound for the encoding bit rate for lossless data compression is the source information rate, also known as the entropy rate.

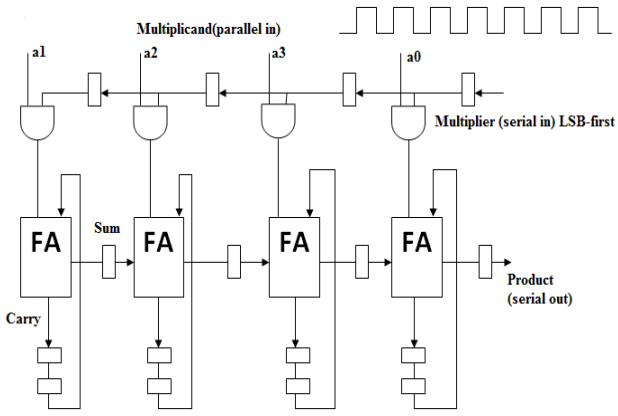

Fig.7. Systolic Design Diagram

E. *Hybrid Type Multiplier*

Hybrid multiplier is another type of multiplier, it provide the combination of both modified booth multiplier and Wallace tree multiplier. The hybrid type multiplier improves the performance at low power. The great timing advantage of the Wallace tree multiplier along with the great power advantage of the bypass scheme in the carry save array multiplier can be combined in mixed multiplier architecture. The 32 bit values can be multiplied by splitting them in two 16 bit. If the first 32bit value is (X, Y) and the second is (W, Z) four 32 bit partial products are generated. A = X x Z, B = Y x Z, C = X x W and D = Y x W. These four partial products are shifted and added together to produce the final 64 bit multiplication result [23].

The main operation behind these operand splitting is to use different multiplier architectures for each different partial multiplication. So, from A = X x Z and C = X x W performance is gained while from B = Y x Z, and D = Y x W power is gained, if half of one or both operands usually contains more no of 0s than 1s, this specific half should be passed through the bypass array multiplier for greater power advantages This multiplier can give good results of power consumption and dynamic power savings but it is not sure give good results in the other parameters such as time and area. Therefore it's hard to implement Wallace tree in FPGA for its irregular and complex interconnection. By using a faster and efficient multiplier which possesses familiar interconnection model than Wallace tree multiplier can give better results than this architecture. The modified booth algorithm provides the better area performance and Wallace tree provides due to reduced delay. However building a regular structure becomes a challenge.

F. Mixed-Style Multiplier

In low power VLSI design, it satisfies Moore's law and it produces the consumer electronic goal with low power consumption and with high battery backup. In this modern world the saving power is an important one. The dynamic power is the dominant property in the technology above 0.1m. The leakage current is more important in a smaller technology. Dynamic power dissipation is occurred while charging the load capacitance in a circuit. By adding the transient power consumption (PT) and capacitive-load power consumption (DL) it results in the dynamic power consumption of CMOS IC is calculated. For the past few years in order to minimize the dynamic power dissipation in arithmetic circuits especially in digital multipliers [28], [29] the different technology are applied in the architecture.

In this mixed style multiplier, it consists of two pants, one is array pant and another one is tree pant. The array pant is given as carry save array multiplier it combined with bypass technique giving good result. To design the low power combinational circuit by using a bypassing logic blocks technology, their function is not required. The bypass technology is given by the low delay and area overhead component like transmission gates. The switching activity of the circuit avoiding and which produces dynamic power saving. The bypassing technique is done in the array pant of the multiplier.

Contrary to GF (2), a number of different representations are commonly used for the elements of a finite field GF (2m). The simplest representation is in polynomial basis, where the multiplication involves multiplying the two polynomials (carry-free coefficient multiplication) and then finding the residue modulo a given irreducible polynomial p (t). In general, the reduction modulo an irreducible polynomial requires polynomial division. However, the polynomial division is very costly to implement in hardware. For an efficient implementation, it is necessary to find a method to perform the field multiplication without division. Two kinds of multiplier are available in the mixed style multiplier, there are array part and tree part multiplier. When we choose the array part carry save array multiplier will provide better result when it's combined with the bypassing technique. By using bypassing technique the design of low power combinational circuit is introduced but their function is not required. This bypassing is using low delay and area overhead component. This technique is avoiding switching activity in the circuit and saving more dynamic power. This kind of multiplier has linear delay circuit [22]. This bypassing technique is applied into the array part of the multiplier

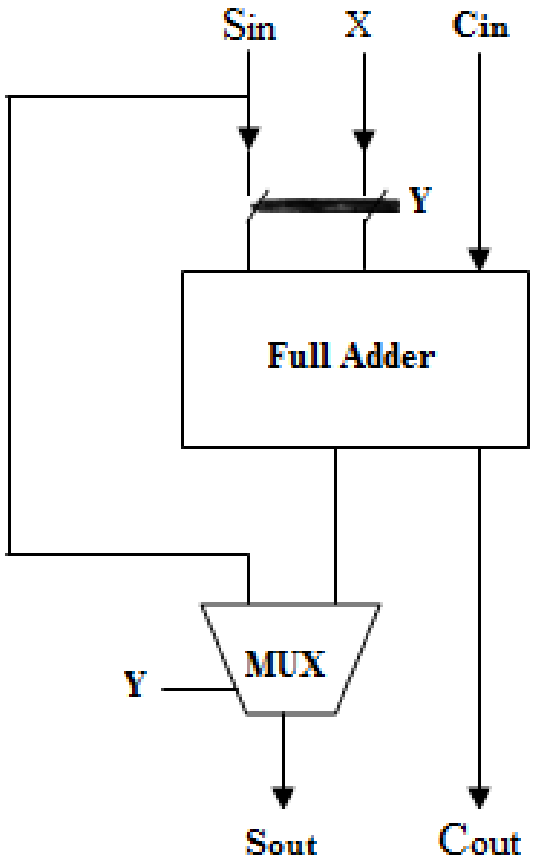

Fig.8. The FAB Cell

With advances in technology, many scientist have tried and trying to design multipliers which offer high speed, low power consumption, layout regularity and hence less area or even combination of them in multiplier. This multiplier can give good results of power consumption and dynamic power savings but it is not sure give good results in the other parameters such as time and area. Therefore it's hard to implement Wallace tree in FPGA for its irregular and complex interconnection. By using a faster and efficient multiplier which possesses familiar interconnection model than Wallace tree multiplier a give better result than this architecture.

The great timing advantage of the Wallace tree multiplier along with the great power advantage of the bypass scheme in the carry save array multiplier can be combined in mixed multiplier architecture. The 32 bit values can be multiplied by splitting them in two 16 bit. If the first 32bit value is (X, Y) and the second is (W, Z) four 32 bit partial products are generated. A = X x Z, B = Y x Z, C = X x W and D = Y x W. These four partial products are shifted and added together to produce the final 64 bit multiplication result [23]. In bypassing technique the array multiplier are choose, since it has regular interconnection, which help to skip the unwanted blocks. The array multipliers have linear delay circuit [30], [31]. The main operation behind this operand splitting is to use different multiplier architectures for each different partial multiplication. So, from A = X x Z and C = X x W performance is gained while from B = Y x Z, and D = Y x W power is gained, if half of one or both operands usually contains more no of 0s than 1s, this specific half should be passed through the bypass array multiplier for greater power advantages

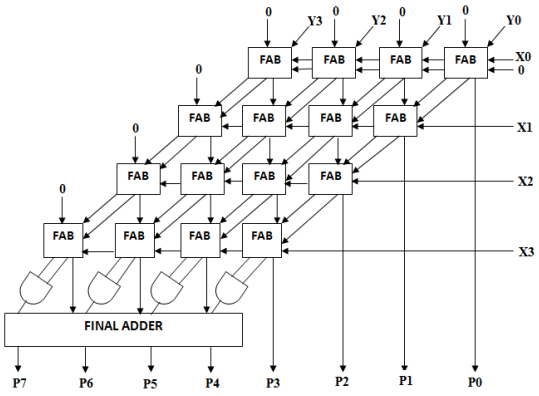

Fig. 9: The carry-save array multiplier with bypass

X= (Xn-1, X1, X0) and Y= (Ym-1, Y1, Y0) are the functionality of carry save array multiplier are fed in to FAB cell. When y=0 the multiplexer propagate the Sin and Sout. At this time the transmission gate in FAB cell locked.Input of full adder to prevent certain operation. When y=1 the full adder work and produce Sout. Another one is tree part multiplier. Logarithmic circuit delay and produce results in faster way is the advantage of the tree part multiplier The SPST adder and carry save array multiplier can be combined in mixed multiplier architecture

The power consumption and dynamic power saving this multiplier give good result. But time and area it is not sure give good result. It's irregular and complex interconnection

it's hard to implement Wallace tree in FPGA. By using faster and efficient multiplier it give better result.

G. Shift and Add Multiplier

Shift-and-add multiplication is similar to the multiplication performed by paper and pencil. This method adds the multiplicand X to itself Y times, where Y denotes the multiplier. To multiply two numbers by paper and pencil, the algorithm is to take the digits of the multiplier one at a time from right to left, multiplying the multiplicand by a single digit of the multiplier and placing the intermediate product in the appropriate positions to the left of the earlier results A standard approach that might be taken by a novice to perform multiplication is to "shift and add", or normal "long multiplication". The original algorithm shifts the multiplicand left with zeros inserted in the new positions, so the least significant bits of the product cannot change after they are formed. Instead of shifting the multiplicand left, we can shift the product to the right. Therefore the multiplicand is fixed relative to the product, and since we are adding only n bits, the adder needs to be only n bits wide. Only the left half of the 2n-bit product register is changed during the addition. Another observation is that the product register has an empty space with the size equal to that of the multiplier. As the empty space in the product register disappears. In consequence, the final version of the multiplier circuit combines the product (A register) with the multiplier (Q register). The A register is only n bits wide, and the product is formed in the A and Q registers. That is, for each column in the multiplier, shift the multiplicand the appropriate number of columns and multiply it by the value of the digit in that column of the multiplier, to obtain a partial product. The partial products are then added to obtain the final result:

```
  0 0 1 0 1 1
  0 1 0 0 1 1
  ---------
  0 0 1 0 1 1
0 0 1 0 1 1
0 0 0 0 0 0
0 0 0 0 0 0
0 0 1 0 1 1
-----------
0 0 1 1 0 1 0 0 0 1
```

Fig. 10: Basic Multiplication Process

With this system, the number of partial products is exactly the number of columns in the multiplier. Shift-and-add multiplication is similar to the multiplication performed by paper and pencil. This method adds the multiplicand 'X' to itself 'Y' times, where 'Y' denotes the multiplier. To multiply two numbers by paper and pencil, the algorithm is to take the digits of the multiplier one at a time from right to left, multiplying the multiplicand by a single digit of the multiplier and placing the intermediate product in the appropriate positions to the left of the earlier results.

To perform the entire operations for getting the final product, the conventional architecture for shift and add multipliers require many switching activities. So the dynamic power dissipation is more in conventional architecture. By eliminating or reducing the sources switching activity in the conventional multiplier, low power architecture of multiplier can be derived. Being one among the functional components of many digital systems the reduction of power dissipation in multipliers should be as much as possible [35]. It is possible to reduce the number of partial products by half, by using the technique of radix 4 Booth recoding. So the dynamic power dissipation is more in conventional architecture.

Being one among the functional components of many digital systems the reduction of power dissipation in multipliers should be as much as possible. Many research efforts have been devoted for reducing the power dissipation of different multipliers. Among multipliers, tree multipliers are used in high speed applications such as filters, but these require large area. The carry-select-adder (CSA)-based radix multipliers, which have lower area overhead, employ a greater number of active transistors for the multiplication operation and hence consume more power. Among other multipliers, shift-and-add multipliers have been used in many applications for their simplicity and relatively small area requirement [46].

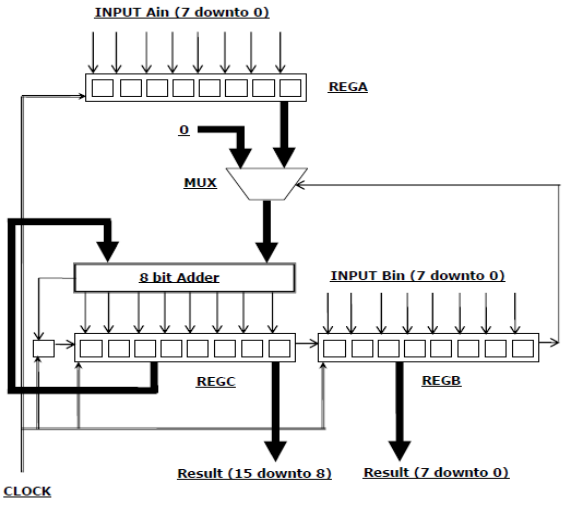

Fig. 11. Shift Add Multiplier Design Implementation

H. Modified BZ-FAD Multiplier

To derive low-power architecture, we concentrate our effort on eliminating or reducing the sources of the switching activity discussed in the previous section. The proposed architecture is called BZ-FAD. A ring counter is used to select B (n) in the nth cycle. As will be seen later, the same counter can be used for block M2 as well. The ring counter used in the proposed multiplier is noticeably wider (32 bits vs. 5 bits for a

32-bit multiplier) than the binary counter used in the conventional architecture; therefore an ordinary ring counter, if used in BZ-FAD, would raise more transitions than its binary counterpart in the conventional architecture. To minimize the switching activity of the counter, we utilize the low-power ring counter.

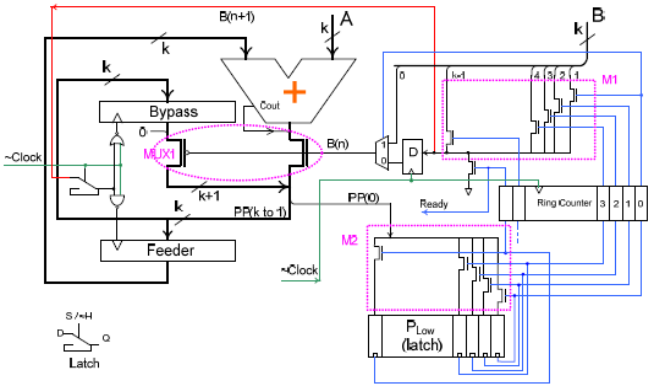

Fig.12. Modified BZ FAD Booth Encoder

In brief, from the six sources of activity in the multiplier, we have eliminated the shift of the B register, reduced the activities of the right input of the adder, and lowered the activities on the multiplexer select line. In addition, we have minimized the activities in the adder, the activities in the counter, and the shifts in the PP (partial product) register. The proposed architecture, however, introduces new sources of activities. These include the activities of a new multiplexer which has the same size as that of the multiplexer of the conventional architecture. Note that the higher part of the partial product in both architectures has the same activity. As will be seen in Section I V the net effect is a lower switching activity for BZ-FAD compared to that of the conventional multiplier. Shift of the PP Register In the conventional architecture, the partial product is shifted in each cycle giving rise to transitions.

Inspecting the multiplication algorithm reveals that the multiplication may be completed by processing the most significant bits of the partial product, and hence, it is not necessary for the least significant bits of the partial product to be shifted. We take advantage of this observation in the BZ-FAD architecture. Notice that in above Figure 14 for P Low, the lower half of the partial product, we use k latches (for a k-bit multiplier). These latches are indicated by the dotted rectangle M2 in the above Figure. To evaluate the efficiency of the proposed architecture, we implemented three different Radix-2 16-bit multipliers corresponding to the conventional, BZ-FAD and SPST architectures. The SPST (Spurious Power Suppression Technique) architecture is a very low-power tree-based array multiplier published recently in the literature [36]. In general, array multipliers offer high speed and low power consumption. However they occupy a lot of silicon area. The SPST results presented in this paper are based on our implementation of this multiplier and are in agreement with the authors' published results. For SPST implementation we used the circuit details of [36], [37] and [38]. To determine the effectiveness of the power reduction techniques discussed in section II, we have reported in Table I the switching activities of major common blocks of the BZFAD and conventional multipliers. As an example, the adder in BZ-FAD has 38.16% less switching activity compared to that of the conventional architecture. The higher switching activity of the BZ-FAD multiplexer is due to its higher fan-out [47].

Statistical Technique

There are number of techniques such as array multiplier, Booth multiplier, Wallace tree multiplier, mixed style multiplier, hybrid multiplier, Bit-serial multiplier, and twin-piped serial – parallel multiplier, all it's are perform binary multiplication. It is depends on the factors such as latency, throughput, area, and design complexity. More efficiency parallel approach use some sort of array pant or tree pant of full adder partial products. Wallace tree has been used in this project in order to accelerate multiplication by compressing the number of partial products.

This design is done using half adders; Carry save adders and the Carry Look Ahead adders to speed up the multiplication. This review discusses the alternating direction method of multipliers (ADMM), a simple but powerful algorithm that is well suited to distributed convex optimization, and in particular to problems arising in applied statistics and machine learning. It takes the form of a decomposition-coordination procedure, in which the solutions to small local sub problems are coordinated to find a solution to a large global problem.

COMPARISON TABLE FOR STATISTICAL TECHNIQUES:

Multiplier Topology	Maximum Delay D(ns)	Power at mw (at higher speed)	Area total CLB's(#)
Hybrid multiplier [42]	8.413ns	288.47uw	5096
Array multiplier [43]	217.8	154	494
Booth multiplier [43]	39.69	144	7496
Wallace tree multiplier [44]	16.235	62.91	444
Bit-Serial Multiplier[45]	123.52	25.891	1439
Mixed style multiplier [45]	143.21	20.581	1328
Shift and Add Multiplier [46]	95	295	2,911 (µm2) [47]
Modified BZ-FAD Multiplier [46]	61	271	3,903 (µm2) [47]

ADMM can be viewed as an attempt to blend the benefits of dual decomposition and augmented Lagrangian methods for constrained optimization, two earlier approaches that we review in §2. It turns out to be equivalent or closely related to many other algorithms as well, such as Douglas-Rachford splitting from numerical analysis, Spingarn's method of partial inverses, Dykstra's alternating projections method, Bregman iterative algorithms for 1 problems in signal processing, proximal methods, and many others. The fact that it has been re-invented in different fields over the decades

underscores the intuitive appeal of the approach. It is worth emphasizing that the algorithm itself is not new, and that we do not present any new theoretical results. It was first introduced in the mid-1970s by Gabay, Mercier, Glowinski, and Marrocco, though similar ideas emerged as early as the mid-1950s. As shown in the figure 12.

Since there are four sign extension values generated namely sign 1E, 2E, 3E and 4E for the partial product PP1, PP2, PP3 and PP4 respectively. The arrangement of total four partial product s is shown in the figure below. The second partial product had to be shifted left by two bits before adding to the first partial product. Hence the third will be shifted left by four where as for fourth it will be shifted left by six. In the hardware implementation Array multiplier, Booth multiplier and Wallace tree multiplier, which are suitable for VLSI Implementation at CMOS level.

Summary and Conclusion

In modern VLSI design the power estimation tools are required to manage the power consumption. In the design phase, it avoid a costly redesign process the average power dissipation, it would not make sense to expect to estimate power without some information about the circuit input patterns. we would to do in order to qualify a chip with a certain power rating that is expected type hold irrespective of the application, we have presented a number of power estimation techniques it provide strong pattern dependence problem, by using many multiplier such as array multiplier, booth multiplier, Wallace tree multiplier, mixed style multiplier, hybrid multiplier, bit-serial multiplier, twin-piped serial-parallel multiplier it give good result for operands which give greater number of bits. Compare to other multiplier, array multiplier requires more power consumption and optimum number of components required, but the delay for this multiplier is larger than the Wallace tree multiplier. The booth multipliers require low power requirement and les delay requirement. The main advantage of low power VLSI design is power saving with great battery backup.

References

[1] Addanki Purna Ramesh1, Dr.A.V. N. Tilak2 and Dr.A.M.Prasad3" EFFICIENT Implementation Of 16-Bit Multiplier-Accumulator Using Radix-2 Modified Booth Algorithm And Spst Adder Using Verilog" International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.3, June 2012

[2] R. W. Brodersen, A. Chandrakasan, S. Sheng, Technologies for personal communications," 1991 Symp. on VLSI circuits, Tokyo, Japan, pp. 5-9, 1991.

[3] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," IEEE Journal of Solid-State Circuits, vol. 27, no. 4, pp. 473-484, April 1992.

[4] WorkshopWorking Group Reports, Semiconductor Industry Association, pp. 22-23, Nov. 17-19, 1992, Irving, Texas.

[5] S. Chowdhury and J. S. Barkatullah, "Estimation of maximum currents in MOS IC logic circuits," IEEE Transactions on Computer-Aided Design, vol. 9, no. 6, pp. 642-654, June 1990.

[6] A. D. Booth, "A signed binary multiplication technique," Quart. J. Math., vol. IV, pp. 236–240, 1952.

[7] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.

[8] N. R. Shanbag and P. Juneja, "Parallel implementation of a 4x4-bit multiplier using modified Booth's algorithm," IEEE J. Solid-State Circuits, vol. 23, no. 4, pp. 1010–1013, Aug. 1988.

[9] A. R. Cooper, "Parallel architecture modified Booth multiplier," Proc. Inst. Electr. Eng. G, vol. 135, pp. 125–128, 1988.

[10] K.H.Chen and Y.S.Chu, "A low power multiplier with spurious power suppression technique",IEEE Trans.Very Large Scale Integr.(VLSI)Syst., Vol.15, no-7,pp846-850, July 2007.

[11] J. Choi, J. Jeon, and K. Choi, "Power minimization of functional units by partially guarded computation," in proc. IEEE Int. Symp. Low Power Electron. des., 2000, pp. 131–136.

[12] O. Chen, S.Wang, and Y. W.Wu, "Minimization of switching activitiesof partial products for designing low-power multipliers," IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 3, pp. 418–433, Jun.2003.

[13] S. Chowdhury and J. S. Barkatullah, "Estimation of maximum currents in MOS IC logic circuits," IEEE Transactions on Computer-Aided Design, vol. 9, no. 6, pp. 642-654, June 1990.

[14] S. Devadas, K. Keutzer, and J. White, "Estimation of power dissipation in CMOS combinational circuits using Boolean function manipulation," IEEE Transactions on Computer-Aided Design, vol. 11, no. 3, pp. 373-383, March 1992.

[15] H. Kriplani, F. Najm, and I. Hajj, "Maximum current estimation in CMOS circuits," 29th ACM/IEEE Design Automation Conference, Anaheim, CA, pp. 2-7, June 8-12, 1992.

[16] www.wikipedia.com for Booth multiplication algorithm

[17] "A Novel Parallel Multiply and Accumulate (V-MAC) Architecture Based On Ancient Indian Vedic Mathematics" Himanshu Thapliyal and Hamid RArbania.

[18] Morris Mano, "Computer System Architecture",PP. 346-347, 3rd edition,PHI. 1993.

[19] Jorn Stohmann Erich Barke, "A Universal Pezaris ArrayMultiplier Generator for SRAM-Based FPGAs" IMS-Institute of Microelectronics System, University of Hanover Callinstr, 34,D-30167 Hanover,Germany.

[20] Moises E. Robinson and Ear Swartzlander, Jr."A Reduction Scheme to Optimize the Wallace Multiplier" Department of Electrical and Computer Engineering, University of Texas at Austin, USA.

[21] Tam Anh Chu, "Booth Multiplier with Low Power High Performance Input Circuitary", US Patent, 6.393.454 B1,May 21, 2002.

[22] S. M. Kang, "Accurate simulation of power dissipation in VLSI circuits," IEEE Journal of Solid-State Circuits, vol. SC-21, no. 5, pp. 889-891, Oct. 1986.

[23] G. Y. Yacoub and W. H. Ku, "An accurate simulation technique for short-circuit power dissipation based on current component isolation," IEEE International Symposium on Circuits and Systems, pp. 1157-1161, 1989.

[24] A-C. Deng, Y-C. Shiau, and K-H. Loh, "Time domain current waveform simulation of CMOS circuits," IEEE International Conference on Computer-Aided Design, Santa Clara, CA, pp. 208-211, Nov. 7-10, 1988.

[25] F. Dresig, Ph. Lanches, O. Rettig, and U. G. Baitinger, "Simulation and reduction of CMOS power dissipation at logic level," European Design Automation Conference, pp. 341-346, 1993.

[26] Bodasingi Vijay, Bhaskar Valiveti Ravi, Tejesvi Reddi,Surya Prakash Rao" Implementation of Radix-4 Multiplier with a Parallel MAC unit using MBE Algorithm" International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 5, July 2012

[27] "ASIC Implementation of 4 Bit Multipliers" Pravinkumar Parate, IEEE Computer society. ICETET, 2008.25.

[28] M. Karlsson, "A generalized carry-save adder array for digital signal is processing," in 4th Nordic Signal Processing Symposium. IEEE, 2000, pp. 287–290.

[29] S. M. Lee, J. H. Chung, H. S. Yoon, and M. M. O. Lee, "High speed and ultra low power 16x16 mac design using tg techniques for webbased multimedia system," in Asia South Pacific Design Automation Conference. ACM/IEEE, 2000, pp. 17–18.

[30] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective – Third Edition. Addison-Wesley, 2004.

[31] M. Jeevitha1, R.Muthaiah2, P.Swaminathan3review Article: Efficient Multiplier Architecture In Vlsi Design" P.G. Scholar, School Of Computing, Sastra University, Tamilnadu, India 2assoc. Prof., School Of Computing, Sastra University, Tamilnadu, India 3dean, School Of Computing, Sastra University, Tamilnadu, India.. Vol. 38 No.2 Edu30th April 2012

[32] Navdeep Goel and Lalit Garg "Comparative Analysis of 4-bit CMOS Multipliers" International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011 Proceedings published by International Journal of Computer Applications® (IJCA)

[33] H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buer circuits," IEEE Journal of Solid-State Circuits, vol. SC-19, no. 4, pp. 468-473, Aug. 1984.

[34] Muhammad H. Rais, member IEEE" Efficient Hardware Realization of Truncated Multipliers using FPGA" International Journal of Engineering and Applied Sciences 5:2 2009.

[35] C. N.Marimuthu1, Dr. P. Thangaraj2, Aswathy Ramesan3" Low Power Shift and Add Multiplier Design" International Journal of Computer Science and Information Technology, Volume 2, Number 3, June 2010

[36] Kuan-Hung Chen and Yuan-Sun Chu, "A Low-Power Multiplier With the Spurious Power Suppression Technique," IEEE Trans. On Very Large Scale Integration (VLSI) Systems, vol. 15, no. 7, July 2007, pp. 846-850.

[37] K. H. Chen, K. C. Chao, J. I. Guo, J. S. Wang, and Y. S. Chu, "An efficient spurious power suppression technique (SPST) and its applications on MPEG-4 AVC/H.264 transform coding design," in Proc. IEEE Int. Symp. Low Power Electron. Des., 2005, pp. 155–160.

[38] K. H. Chen, Y. M. Chen, and Y. S. Chu, "A versatile multimedia functional unit design using the spurious power suppression technique," in Proc. IEEE Asian Solid-State Circuits Conf., 2006, pp.111–114.

[39] C. S. Wallace, "A suggestion for fast multipliers," IEEE Trans. Electron. Computers, vol. 13, pp. 14-17, Feb.1964.

[40] H. Lee, "A power-aware scalable pipelined Booth multiplier" in Proc. of IEEE Int. SOC Conf., 2004, pp. 123–126.

[41] http://en.wikipedia.org/w/index.php?search=bit+serial+multiplier&title=Special%A3Search

[42] Athira Koranath, Sonali Agrawal "Comparison of different multiplier algorithms and 1D-DWT as an application" IOSR Journal of Electronics and Communication Engineering (IOSRJECE)ISSN : 2278-2834 Volume 1, Issue 1 (May-June 2012)

[43] Addanki Purna Ramesh1, Dr.A.V. N. Tilak2 and Dr.A.M.Prasad3 "Efficient Implementation Of 16-Bit Multiplier-Accumulator Using RADIX-2 Modified Booth Algorithm and SPST Adder Using Verilog "International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.3, June 2012

[44] http://www.doe.carleton.ca/~shams/ELEC5705Y/MantoPresentation.ppt

[45] R. Dutta "Power Efficient VLSI Architecture for IIR Filter using Modified Booth Algorithm" International Journal of Advanced Research in Technology Vol. 2 Issue 1, Jan 2012

[46] 4.K.N. Vijeyakumar, V. Sumathy, Sriram Komanduri and C. Chrisjin Gnana Suji "Design of Low-Power High-Speed Error Tolerant Shift and Add Multiplier" Journal of Computer Science 7 (12): 1839-1845, 2011.

[47] M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram "BZ-FAD: A Low-Power Low-Area Multiplier based on Shift-and-Add Architecture"To appear in IEEE Trans. on VLSI Systems, 2008