

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

The design of high-speed and low-power VLSI architectures need efficient arithmetic processing units, which are optimized for the performance parameters, namely, speed and power consumption. Adders are the key components in general purpose microprocessors and digital signal processors. They also find use in many other functions such as subtraction, multiplication and division. As a result, it is very pertinent that its performance augers well for their speed performance. Furthermore, for the applications such as the RISC processor design, where single cycle execution of instructions is the key measure of performance of the circuits, use of an efficient adder circuit becomes necessary, to realize efficient system performance. Additionally, the area is an essential factor which is to be taken into account in the design of fast adders. Towards this end, high-speed, low power and area efficient addition and multiplication have always been a fundamental requirement of high-performance processors and systems. The major speed limitation of adders arises from the huge carry propagation delay encountered in the conventional adder circuits, such as ripple carry adder and carry save adder.

Power dissipation is one of the most important design objectives in integrated circuit, after speed. Digital signal processing (DSP) circuits whose main building block is a Multiplier-Accumulator (MAC) unit. High speed and low power MAC unit is desirable for any DSP processor. This is because

speed and throughput rate are always the concerns of DSP system. Due to rapid growth of portable electronic systems like laptop, calculator, mobile etc., and the low power devices have become very important in today world. Low power and high-throughput circuitry design are playing the challenging role for VLSI designer. For real-time signal processing, a high speed and high throughput MAC unit is always a key to achieve a high performance digital signal processing system. A regular MAC unit consists of multipliers and accumulators that contain the sum of the previous consecutive products. The main motivation of this work is to investigate various multiplier and adder architectures which are suitable for implementing Low power, area efficient and high speed MAC unit.

This chapter begins with the basic building blocks used for addition and multiplication, and it go through different researcher's survey on adders, multipliers and MAC unit.

2.2 BASIC ADDER BLOCKS

2.2.1 Half Adder

The Half Adder (HA) is the most basic adder. It takes in two bits of the same weight, and creates a sum and a carryout. If the two inputs a and b have a weight of 2^i (where i is an integer), sum has a weight of 2^i , and carryout has a weight of $2^{(i+1)}$. Equations 2.1 and 2.2 are the Boolean equations for sum and carryout, respectively,

$$\text{Sum} = a \oplus b \quad (2.1)$$

$$\text{Carry} = a.b \quad (2.2)$$

2.2.2 Full Adder

The Full Adder (FA) is useful for additions that have multiple bits in each of its operands. It takes three inputs and creates two outputs, a sum and a carryout. The inputs have the same weight, 2^i , the sum output has a weight of 2^i , and the carryout output has a weight of $2^{(i+1)}$. The FA differs from the HA and it has a $carry_{in}$ as one of its inputs, allowing for the cascading of this structure. Equations 2.3 and 2.4 are the Boolean equations for the FA sum and FA carryout, respectively. In both these equations c_{in} means $carry_{in}$.

$$Sum_i = a \oplus b \oplus c_{in_i} \quad (2.3)$$

$$Carry_i = a_i \cdot b_i + b_i \cdot c_{in_i} + a_i \cdot c_{in_i} \quad (2.4)$$

2.2.3 Partial Full Adder

The Partial Full Adder (PFA) is a structure that implements intermediate signals that can be used in the calculation of the carry bit. It is an extension of FA which include the signals generate (g), kill (k), and propagate (p). When $g=1$, it means carryout will be 1 (generated) regardless of $carry_{in}$. When $k=1$, it means carryout will be 0 (killed) regardless of $carry_{in}$. When $p=1$, it means carryout will equal $carry_{in}$ ($carry_{in}$ will be propagated). Table 2.1 reflects these three additional signals, with a comment on the carryout bit in an additional column. Equations 2.5– 2.7 are the Boolean equations for generate, kill, and propagate, respectively. It should be noted for the propagate signal, the XOR function can also be used, since in the case of $a, b=1$, the generate signal will assert that carryout is 1. The Boolean equations for the sum and carryout can be written as functions of g, p, or k shown by Equations 2.8 and 2.9. Figure 2.1 shows a circuit for creating the generate signal, propagate signal, and sum signal. It is a partial full adder because it does not calculate the carryout signal directly; rather, it creates the signals needed to calculate the carryout signal.

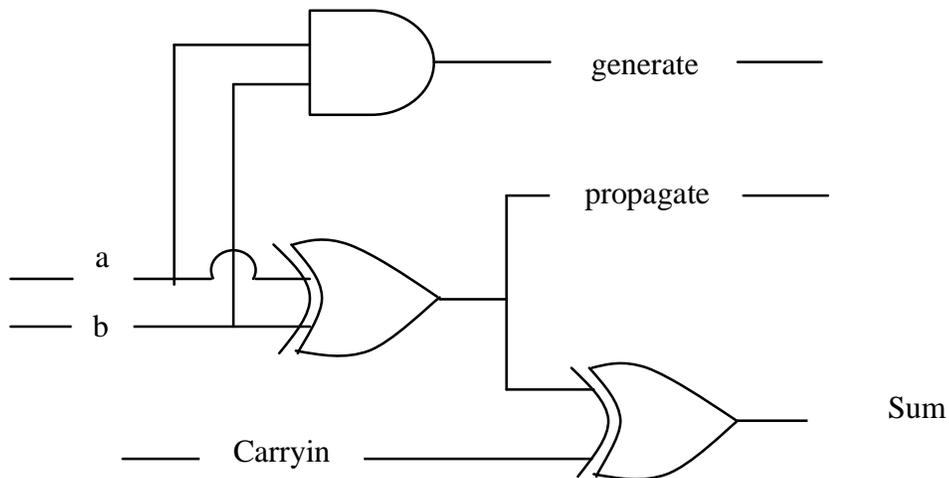


Figure 2.1: Gate level schematic for Partial Full Adder (PFA)

$$\text{Generate}_i(g_i) = a_i \cdot b_i \quad (2.5)$$

$$\text{Kill}_i = a_i \cdot b_i \quad (2.6)$$

$$\text{Propagate}_i(p_i) = a_i + b_i = a_i \oplus b_i \quad (2.7)$$

$$\text{sum}_i = p_i \oplus \text{cin}_i \quad (2.8)$$

$$\text{carryout}_{i+1} = a_i \cdot b_i + b_i \cdot \text{carryin}_i + a_i \cdot \text{carryin}_i \quad (2.9)$$

Table 2.1 Extended truth table for 1-bit adder

Inputs			Outputs					
Carryin	a	b	Carryout	Sum	g	k	p	Carry Status
0	0	0	0	0	0	1	0	Delete
0	0	1	0	1	0	0	1	Propagate
0	1	0	0	1	0	0	1	Propagate
0	1	1	1	0	1	0	1	Generate/propagate
1	0	0	0	1	0	1	0	Delete
1	0	1	1	0	0	0	1	Propagate
1	1	0	1	0	0	0	1	Propagate
1	1	1	1	1	1	0	1	Generate/propagate

2.3 BASICS OF MULTIPLIERS

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of following – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier, thus making them suitable for various high speed, low power, and compact VLSI implementation. The common multiplication method is adding and shift algorithm. Multiplication is a mathematical operation at its simplest is an abbreviated process of adding an integer to itself, a specified number of times. A number (multiplicand) is added to itself a number of times as specified by another number (multiplier) to form result (product). Multiplication hardware often consumes much time and area compared to other arithmetic operations. Digital signal processors use a multiplier/MAC unit as a basic building block and the algorithms they run are often multiply-intensive. Multiplication-based operations such as Multiply and Accumulate (MAC) are currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its arithmetic and logic unit. This chapter discusses different architectures for multiplication and the methods that improve speed and/or area. Also, it is important to consider these methods in the context of VLSI design. It is beneficial to find structures that are modular and easy to layout. Many of the architectures described in this chapter will be used in the implementation of the multiply-accumulate unit

2.3.1 Array Multiplier

Each multiplicand is multiplied by a bit in the multiplier, generating N partial products. Each of these partial products is either the multiplicand shifted by some amount or 0. This is illustrated in Figure 2.2 for an $M \times N$

multiply operation. This figure can map directly into hardware and is called the array multiplier. The generation of partial products consists of simple AND'ing of the multiplier and the multiplicand. The accumulation of these partial products can be done with rows of ripple adders. Thus, the carry out from the least significant bit ripples to the most significant bit of the same row, and then down the "left side" of the structure. The partial products are added in ripple fashion with half and full adders. A full adder's inputs require the carry_{in} from the adjacent full adder in its row and the sum from a full adder in the above row. Rao et al (2009) states that finding the critical path in this structure is non-trivial, but once identified, results in multiple critical paths. It requires a lot of time to optimize the adders in the array since all adders in the multiple critical paths need to be optimized to result in any speed increase (this implies optimization of both the sum and carryout signals in a full adder). The delay basically comes down to a ripple delay through a row, and then down a column, so it is linearly proportional ($t_d \sim M+N$) to the sum of the sizes of the input operands. Conventional linear array multipliers consist of rows of carry-save adders (CSA). In a linear array multiplier, the data propagates down through the array, each row of CSA's adds one additional partial-product to the partial sum. Since the intermediate partial sum is kept in a redundant, there is no carry propagation in carry save adder. This means that the delay of an array multiplier is only dependent upon the depth of the array, and is independent of the partial-product width. Their high performance and regular structure have perpetuated the use of array multipliers for VLSI math co-processors and special purpose DSP chips. The main disadvantage of the array multiplier is the worst case delay of the multiplier proportional to the width of the multiplier. As operand sizes increase, linear arrays grow in size at a rate equal to the square of the operand size. This is because the number of rows in the array is equal to the length of the multiplier, with the width of each row equal to the width of multiplicand. The large size of full arrays typically prohibits their use, except for small

operand sizes, or on special purpose math chips where a major portion of the silicon area can be assigned to the multiplier array. Another problem with array multipliers is that the hardware is underutilized. As the sum is propagated down through the array, each row of CSA's computes a result only once, when the active computation front passes that row. Thus, the hardware is doing useful work only a very small percentage of the time. This low hardware utilization in conventional linear array multipliers makes performance gains possible through increased efficiency. For example, by overlapping calculations pipelining can achieve a large gain in throughput.

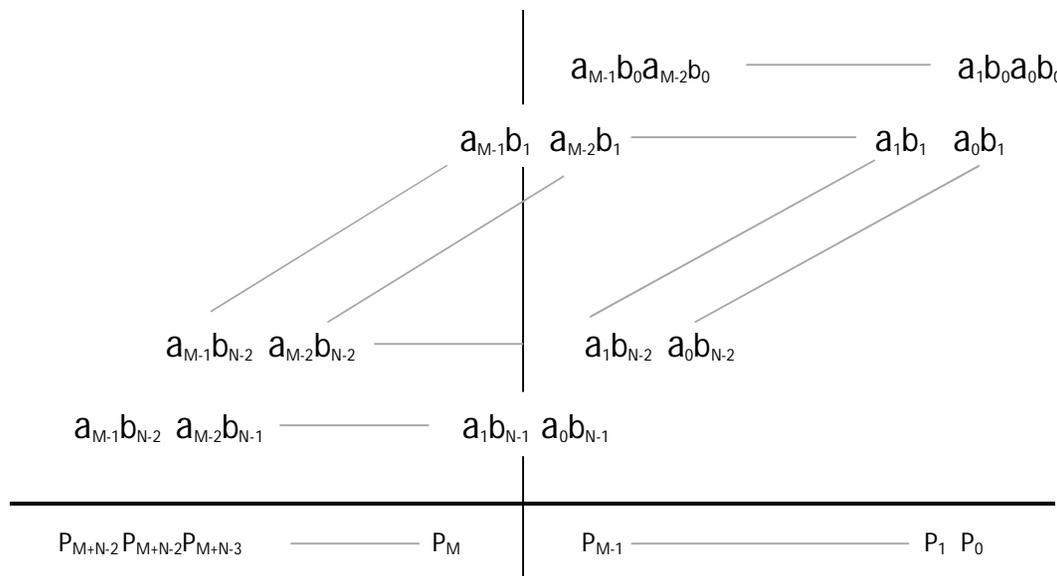


Figure 2.2 Partial product array of an $M \times N$ multiplier

2.3.2 Tree Multiplier

The tree multiplier reduces the time for the accumulation of partial products by adding all of them in parallel, whereas the array multiplier adds each partial product in series. The tree multiplier commonly uses CSAs to accumulate the partial products.

2.3.2.1 Wallace tree

The reduction of partial products using full adders as carry-save adders (also called 3:2 counters) became generally known as the “Wallace Tree”. This architecture reduces the partial products at a rate of $\log_{3/2}N/2$. Figure 2.3 shows an example of tree reduction for an 8×8 -bit partial product tree. The ovals around the dots represent either a full adder (for three circled dots) or a half adder (for two circled dots). This tree is reduced to two rows for a carry-propagate adder after four stages. There are many ways to reduce this tree with CSAs, and this example is just one of them. By using carry-save adder the need of carry propagation in the adder is avoided and latency of one addition is equal to gate delay of adder.

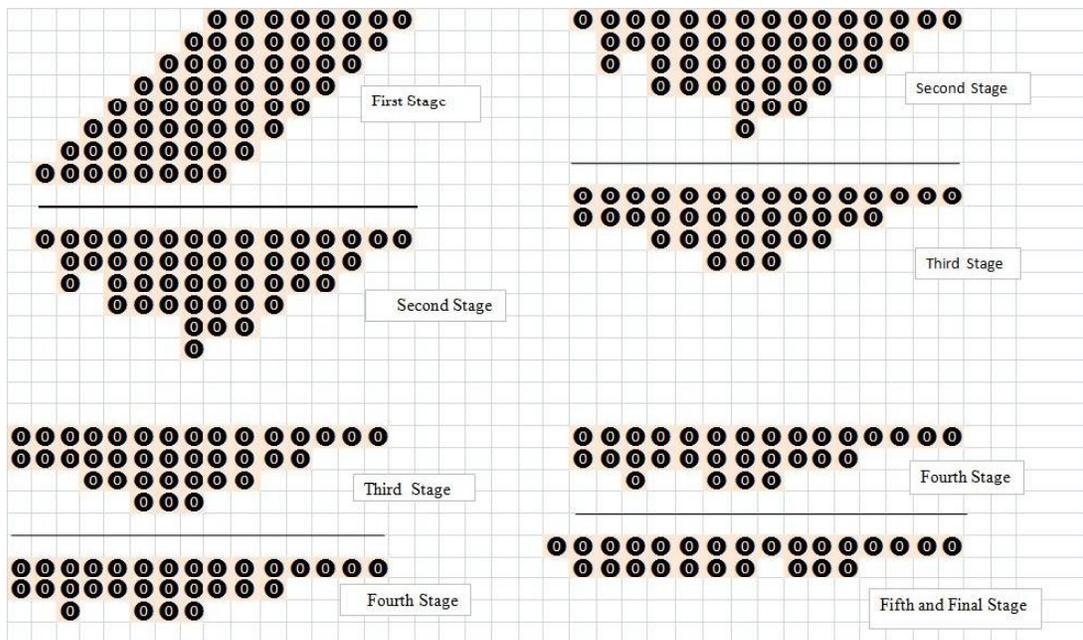


Figure 2.3 Wallace tree for an 8×8 bit partial product tree

2.4 LITERATURE SURVEY

2.4.1 Optimization

Deepak & Kailath (2012) discussed a new multiplier design is proposed which reduces the number of partial products by 25%. This multiplier has been used with different adders available in literature to implement multiplier accumulator (MAC) unit and parameters such as propagation delay, power consumed and area occupied have been compared in each case. From the results, Kogge tone adder has been chosen as it provided optimum values of delay and power dissipation. Later, the results obtained have been compared with that of other multipliers and it has been observed that the proposed multiplier has the lowest propagation delay when compared with Array and Booth multipliers.

2.4.2 Switching Techniques

Chen et al (2003) presented low-power 2's complement multipliers by minimizing the switching activities of partial products using the radix-4 Booth algorithm. Before computation for two input data, the one with a smaller effective dynamic range is processed to generate Booth codes, thereby increasing the probability that the partial products become zero. By employing the dynamic-range determination unit to control input data paths, the multiplier with a column-based adder tree of compressors or counters is designed. To further reduce power consumption, the two multipliers based on row-based and hybrid-based adder trees are realized with operations on effective dynamic ranges of input data. Functional blocks of these two multipliers can preserve their previous input states for non effective dynamic data ranges and thus, reduce the number of their switching operations. It illustrates the proposed multipliers exhibiting low-power dissipation, the theoretical analyzes of switching activities of partial products

are derived. The proposed 16 /spl times/ 16-bit multiplier with the column-based adder tree conserves more than 31.2%, 19.1%, and 33.0% of power consumed by the conventional multiplier. Furthermore, the proposed multipliers with row-based, hybrid-based adder trees reduce power consumption by over 35.3%, 25.3% and 39.6%, and 33.4%, 24.9% and 36.9%, respectively.

Oscal et al (2003) presented low-power 2's complement multipliers by minimizing the switching activities of partial products using the radix-4 Booth algorithm. Before computation for two input data, the one with a smaller effective dynamic range is processed to generate Booth codes, thereby increasing the probability that the partial products become zero. By employing the dynamic-range determination unit to control input data paths, the multiplier with a column-based adder tree of compressors or counters is designed. To further reduce power consumption, the two multipliers based on row-based and hybrid-based adder trees are realized with operations on effective dynamic ranges of input data. Functional blocks of these two multipliers can preserve their previous input states for non effective dynamic data ranges and thus, reduce the number of their switching operations. To illustrate the proposed multipliers exhibiting low-power dissipation, the theoretical analyzes of switching activities of partial products are derived. The proposed 16 /spl times/ 16-bit multiplier with the column-based adder tree conserves more than 31.2%, 19.1%, and 33.0% of power consumed by the conventional multiplier, in applications of the ADPCM audio, G.723.1 speech, and wavelet-based image coders, respectively. Furthermore, the proposed multipliers with row-based, hybrid-based adder trees reduce power consumption by over 35.3%, 25.3% and 39.6%, and 33.4%, 24.9% and 36.9%, respectively. When considering product factors of hardware areas, critical delays and power consumption, the proposed multipliers can outperform the conventional multipliers. Consequently, the multipliers

proposed herein can be broadly used in various media processing to yield low-power consumption at limited hardware cost or little slowing of speed.

Nan-Ying Shen et al (2002) presented Low-power 2's complement multipliers are developed through minimizing switching activities of partial products using the radix-4 Booth algorithm. Before computation, the input datum with the smaller effective dynamic range is processed to generate Booth codes, thereby increasing probabilities of partial products being zero. By employing the dynamic-range determination units to control input data paths, the proposed 16×16 -bit multipliers based on the Yu, Goldovsky, and Mahant-Shetti's low-power approaches are individually implemented. It illustrates the proposed multiplier having low-power dissipation; the theoretical analyses of switching activities of partial products are derived. Compared to the power consumed by the conventional multipliers. The proposed multipliers conserve more than 14%, 30% and 31% of power, respectively.

More & Kshirsagar (2011) presented low power Column bypass multiplier design methodology that inserts more number of zeros in the multiplicand thereby reducing the number of switching activities as well as power consumption. The switching activity of the component used in the design depends on the input bit coefficient. This means if the input bit coefficient is zero, corresponding row or column of adders need not be activated. If multiplicand contains more zeros, higher power reduction can be achieved. To reduce the switching activity is to shut down the idle part of the circuit, which is not in operating condition. Use of look up table is an added feature to this design. Further, low power adder structure reduces the switching activity. Flexibility is another critical requirement that mandates the use of programmable components like FPGAs in such devices.

Saravanan & Madheswaran (2010) analyzed low power high performance Multiply and Accumulate (MAC) unit with Hybrid Encoded Reduced Transition Activity Technique (HERTAT) equipped multiplier and low power 0.13 μ m adder. The developed low power MAC unit is verified for image processing systems exploiting insignificant bits in pixels values and the similarity of neighboring pixels in video streams. The proposed technique reduces dynamic power consumption by analyzing the bit patterns in the input data to reduce switching activities. If the number of 1's less than or equal to three, and applies the proposed encoding technique, otherwise can make use of Booth technique. The proposed adder cell used in the MAC block consumes less power than the other previous adder techniques. This high performance low power MAC can be used in image processing.

Economakos et al (2010) discussed a new technique for the design of combinational circuits for low power is introduced. The basic idea is to bypass blocks of logic when their function is not required, using low delay and area overhead components (transmission gates). The internal state of these blocks is kept unchanged, so the switching activity of the circuit is minimized, resulting to low dynamic power consumption. While this idea offers great savings mainly to array multipliers, due to their regular interconnection scheme, the reduced area and fast speed of tree multipliers is a real temptation for the designer. Therefore, a mixed style architecture, using a traditional, tree based part, combined with a bypass, array based part, is proposed. Through extensive experimentation it has been found that the bypass technique offers minimum power consumption for all cases while the mixed architecture offers a delay \times power product improvement ranging from 1.2x to 6.5x, compared to all other architectures

Rashidi & Pourormazd (2011) presented the methods to reduce dynamic power consumption of a digital Finite Impulse Response (FIR) filter

these methods include low power serial multiplier and serial adder, combinational booth multiplier, shift/add multipliers, folding transformation in linear phase architecture and applied to fir filters to power consumption reduced thus reduce power consumption due to glitching is also reduced. The minimum power achieved is 110mw in fir filter based on shift/add multiplier in 100MHZ to 8staps and 8bits inputs and 8bits coefficients.

Rao et al (2009) Low power multipliers with high clock frequencies play an important role in today's digital signal processing. In this work, the performance analysis of Wallace-tree, Array and Baugh-Wooley multiplier architectures is carried out. Physical verification of all the sub-blocks is performed using HSpice to check their functionality and to optimize for low power by using transistor sizing. Delay and power dissipation of Wallace Tree multiplier is least whereas Array multiplier is best for reduced area applications but not speed. In this work, the area of 5 times 5 Array multiplier is 67.73 times 7 μm^2 is the least when compared to others.

Wen et al (2005) a low power parallel multiplier design, in which some columns in the multiplier array can be turned-off whenever their outputs are known. In this case, the columns are bypassed, and thus the switching power is saved. The advantage of this design is that it maintains the original array structure without introducing extra boundary cells, as did in previous designs. Experimental results show that it saves 10% power for random inputs. Higher power reduction can be achieved if the operands contain more 0s than 1s. Compared with row-bypassing multipliers, this approach achieves higher power reduction with smaller area overhead.

Hwang et al (2007) proposed two novel low power multipliers based on enhanced row bypassing schemes. The essence of the power saving idea is eliminating unnecessary computation via signal bypassing. In an array multiplier, futile computations occur on those columns or rows of adder

corresponding to zero bits in the input operands. Previous designs resort to input gating and output multiplexing to accomplish signal bypassing. The proposed designs, however, successfully resolve the adverse DC power consumption problem due to voltage loss in gated signals and implement the multiplexing mechanism cleverly via clock CMOS (C²MOS) circuitry. Two versions of the design are proposed with one emphasizing on maximizing power saving and the other focusing on reduced circuit complexity. The circuit overheads of both designs are confined to 23.4% and 12.8%, respectively. The proposed designs also achieve better and consistent power saving than previous work under a wide range of V_{dd} and the power saving can be as high as 17%.

Jaina et al (2011) discussed that Real-time signal processing requires high speed and high throughput Multiplier-Accumulator (MAC) unit that consumes low power, which is always a key to achieve a high performance digital signal processing system. In this paper, design of MAC unit is proposed. The multiplier used inside the MAC unit is based on the Sutra "Urdhva Tiryagbhyam" (Vertically and Cross wise) which is one of the Sutras of Vedic mathematics. Vedic mathematics is mainly based on sixteen Sutras and was rediscovered in early twentieth century. In ancient India, this Sutra was traditionally used for decimal number multiplications within less time. The same concept is applied for multiplication of binary numbers to make it useful in the digital hardware. Here, the coding is done in VHDL and synthesis is done in Xilinx ISE series. The combinational delay obtained after synthesis is compared with the performance of the "Modified Booth Wallace Multiplier" and "High speed Vedic multiplier" presented by Ramesh Pushpangadam. They proposed Vedic multiplier seems to have better performance.

2.4.3 Pipelining

Shanthala et al (2009) investigated various pipelined MAC architectures and circuit and the design techniques which are suitable for the implementation of high throughput signal processing algorithms. The goal of this project is to design and VLSI implementation of pipelined MAC for high-speed DSP applications at 180nm technology. For designing the pipelined MAC, various architectures of multipliers and one bit full adders are considered. The static and dynamic one bit full adder was implemented as the basic block. For checking the functionality of the whole system, spice code is written using the HSPICE by defining all the blocks in the circuit as the sub circuits.

To improve the speed of VLSI signal processing systems, a new architecture for a high-speed Multiply-Accumulate (MAC) unit optimized for digital filters is proposed. This unit is designed as a coprocessor for the LEON2 RISC processor. In this work, four parallel MAC units with two dual-port coefficient register files and a three-port general register file and a control unit are included in the co processing block. With the existence of four parallel units, several SIMD format instructions have been added to LEON2 instruction set. Each MAC unit has two 16-bit inputs, 32-bit output register and a programmable Round-Saturate block. The MAC unit uses a new architecture which embeds the accumulate module within the partial products summation tree of the multiplier with minimum overhead. The experimental results demonstrate a high performance in implementation of digital filters at elevated speeds of up to 33 millions of input samples per second.

A 64-bit fixed-point vector multiply-accumulator (MAC) architecture is capable of supporting multiple precisions. The vector MAC can perform one 64×64 , two 32×32 , four 16×16 , or eight 8×8 bit signed/unsigned multiply using essentially the same hardware as a scalar 64-

bit MAC and with only a small increase in delay. The scalar MAC architecture is "vectorized" by inserting mode-dependent multiplexing into the partial product generation and by inserting mode-dependent kills in the carry chain of the reduction tree and the final carry-propagate adder. This is an example of "shared segmentation" in which the existing scalar structure is segmented and then shared between vector modes. The vector MAC is area efficient and can be fully pipelined, which makes it suitable for high-performance processors and, possibly, dynamically reconfigurable processors. The "shared segmentation" method is compared to an alternative method, referred to as the "shared subtree" method, by implementing vector MAC designs using two different technologies and three different vector widths.

Kouretas & Ioannis et al (2013) presented techniques for low-power addition/subtraction in the logarithmic number system (LNS) and quantifies their impact on digital filter VLSI implementation. The impact of partitioning the look-up tables required for LNS addition/subtraction on complexity, performance, and power dissipation of the corresponding circuits is quantified. Two design parameters are exploited to minimize complexity, namely the LNS base and the organization of the LNS word. A round off noise model is used to demonstrate the impact of base and word length on the signal-to-noise ratio of the output of finite impulse response (FIR) filters. In addition, techniques for the low-power implementation of an LNS multiply accumulate (MAC) units are investigated. Furthermore, it is shown that the proposed techniques can be extended to cotransformation-based circuits that employ interpolators. The results are demonstrated by evaluating the power dissipation, complexity and performance of several FIR filter configurations comprising one, two or four MAC units.

Abdelgawad (2013) discussed that Digital Signal Processing (DSP) is one of the capable processing units, but it is not commonly used in WSN because of

the power constraint. The Multiply-Accumulate Unit (MAC) is the main computational kernel in DSP architectures. The MAC unit determines the power and the speed of the overall system; it always lies in the critical path. Developing high speed and low power MAC is crucial to use DSP in the future WSN. In this work, a fast and low power MAC Unit is proposed. The proposed architecture is based on examination of the critical delays and hardware complexities of merged MAC architectures to design a unit with a low critical path delay and low hardware complexity. The new architecture reduces the hardware complexity of the summation network, thus reduces the overall power. Increasing the speed of operation is achieved by feeding the bits of the accumulated operand into the summation tree before the final adder instead of going through the entire summation network. The ASIC implementation of the proposed 32-bit MAC unit saves 5.5% of the area, 9% of the energy, and reduces the delay by 13% compared to the regular merged MAC unit.

Chung-Hsun Huang et al (2002) discussed that look ahead signals to form the multilevel folding architecture for priority-encoding-based designs was used to improve the performance to the order of $O(\log N)$. Analysis showed that both the multilevel look ahead and the multilevel folding techniques could be easily merged and implemented in the dynamic CMOS circuits. For the 256-bit priority encoder, the new design adopting all the proposed techniques can achieve nearly ten times performance while spending nearly half the power consumption as compared to the conventional design, utilizing only a simple look ahead structure. For the 64-bit incrementer/decrementer, the new design adopting all the proposed techniques requires less than one-third delay time as compared to a high-speed carry-select adder (CSA)-based incrementer/decrementer. The power consumption evaluated at the maximum operating frequency and the transistor count of the new incrementer/decrementer are also reduced to 67%

and 35%, respectively, as compared to the CSA-based design. The measurement results indicate that the proposed 256-bit priority encoder and the proposed 64-bit incrementer/decrementer can operate up to 116 and 139 MHz

Chiou-Kou Tung et al (2007) proposed a low-power high-speed CMOS full adder core for embedded system. Based on a new three-input exclusive OR (3-XOR) design, the new hybrid full adder is composed of pass-transistor logic and static CMOS logic. The main design objectives for the full adder core are providing not only low power and high speed but also with driving capability. Using TSMC CMOS 0.35- μm technology, the characteristics of the experimental circuit compared with prior literature show that the new adder improves 1.8% to 35.6% in power consumption, 11.7% to 41.2% in time delay of C_o , and 13.7% to 91.4% in power-delay product of C_o . The circuit is proven to have the minimum power consumption and the fastest response of carry out signal among the adders selected for comparison. Due to the low-power and high-speed properties, both the new exclusive OR circuit and the new full adder can be efficiently integrated in a system-on-a-chip (SoC) or an embedded system.

Vasefi & Fartashand Abid (2005) designed and simulated 4-bit ripple carry adders (RCA), 12-bit carry select adders (CSA), and a 4 times 4 Braun multiplier, based on lowest-number-of-transistor full adders. The designed full adders consist of 10 transistors and were used for n-bit adders with output voltage levels having a maximum of one threshold voltage (V_r) degradation. The 10 transistors adder achieved a 43.68% reduction in the power dissipation compared to the standard CMOS-28T adder. Power consumption can be further reduced by using an extra stack transistor. A 12-transistor adder was also designed for low area array multipliers

Lin & Jin-Fa et al (2006) presented a novel full adder design with only 10 transistors targeting low power arithmetic operation for wireless base band processing. To alleviate the multi threshold voltage loss and the speed degradation problems common in 10T full adder designs, an inverter is successfully embedded along the carry path at no extra circuit overhead. It helps to restore the logic swing and thus the driving capability to enhance the speed of carry propagation. The performance edge becomes even bigger when the adders are cascaded for n-bit ripple carry addition. They further compare their design with other higher transistor count full adder designs. The simulations reveal that their design performs comparably well in speed and power metrics but enjoys the advantage of much lower circuit complexity.

Unsal & Koren (2003) discussed that Power and energy consumption has recently become an important issue and consequently, power-aware techniques are being devised at all levels of system design; from the circuit and device level, to the architectural, compiler, operating system, and networking layers. This paper focuses on power-aware design techniques for real-time systems, while the main focus is on hard real-time, soft real-time systems are considered as well. By start with the motivation for focusing on these systems and provide a brief discussion on power and energy objectives that follows with a survey of current research on a layer-by-layer basis. This paper concludes with illustrative examples and open research challenges and provides an overview of power-aware techniques for the real-time system engineer as well as an up-to-date reference list for the researcher.

Chen et al (2002) designed a power-efficient digital signal processor. This study develops a fundamental arithmetic unit of a low-power adder that operates on effective dynamic data ranges. Before performing an addition operation, the effective dynamic ranges of generate the desired result while the input bits of the unused functional blocks remain in their previous

states. The added result is two input data. Based on a larger effective dynamic range, only selected functional blocks of the adder are activated to then recovered to match the required word length. Using this approach to reduce switching operations of non effective bits, allows input data in 2's complement and sign magnitude representations to have similar switching activities. This investigation proposes a 2's complement adder with two master-stage and slave-stage flip-flops, a dynamic-range determination unit and a sign-extension unit, owing to the easy implementation of addition and subtraction in such system. Furthermore, this adder has a minimum number of transistors addressed by carry-in bits and thus is designed to reduce the power consumption of its unused functional blocks. The dynamic range and sign-extension units are explored in detail to minimize their circuit area and power consumption. Experimental results demonstrate that the proposed 32-bit adder can reduce power dissipation of conventional low-power adders for practical multimedia applications. Besides the ripple adder, the proposed approach can be utilized in other adder cells, such as carry look ahead and carry-select adders, to compromise complexity, speed and power consumption for application-specific integrated circuits and digital signal processors.

Huang & Ercegovic (2005) presented a high-performance low-power design of linear array multipliers based on a combination of the following techniques: signal flow optimization in [3:2] adder array for partial product reduction, left-to-right leapfrog (LRLF) signal flow, and splitting of the reduction array into upper/lower parts. The resulting upper/lower LRLF (ULLRLF) multiplier is compared with tree multipliers. From automatic layout experiments, they find that ULLRLF multipliers have similar power, delay, and area as tree multipliers for $n \leq 32$. With more regularity and inherently shorter interconnects, the ULLRLF structure presents a competitive

alternative to tree structures in the design of fast low-power multipliers implemented in deep submicron VLSI technology.

Kuan-Hung Chen & Yuan-Sun Chu (2007) applied an advanced version of former spurious power suppression technique (SPST) on multipliers for high-speed and low-power purposes. To filter out the useless switching power, there are two approaches, i.e., using registers and using andgates, to assert the data signals of multipliers after the data transition. The SPST has been applied on both the modified Booth decoder and the compression tree of multipliers to enlarge the power reduction. The simulation results show that the SPST implementation with AND gates owns an extremely high flexibility on adjusting the data asserting time which not only facilitates the robustness of SPST but also leads to a 40% speed improvement.

Song & De Micheli (1991) discussed VLSI implementations of high-performance parallel multipliers. Circuit building blocks required for partial-product reduction are analyzed and two schemes leading to highly regular layouts are proposed. The circuit implementations related to the first-scheme in three different BiCMOS technologies are discussed. The die size and performance for nominal design rule values are compared, and the trend in scaling the feature sizes is studied.

Uming Ko et al (2005) implemented high-performance adders, with the objective to optimize performance per watt or energy efficiency as well as silicon area efficiency. The investigation is done using 100 MHz, 32 b carry lookahead (CLA) adders in a 0.6 μm /mCMOS technology, most techniques presented here can also be applied to other parallel adder algorithms such as carry-select adders (CSA) and other energy efficient CMOS circuits. Among the techniques presented here, the double pass-transistor logic (DPL) is found to be the most energy efficient while the

single-rail domino and complementary pass-transistor logic (CPL) result in the best performance and the most area efficient adders, respectively.

Shams & et al (2002) presented a performance analysis of 1-bit full-adder cell. The adder cell is anatomized into smaller modules. The modules are studied and evaluated extensively. Several designs of each of them are developed, prototyped, simulated and analyzed. Twenty different 1-bit full-adder cells are constructed (most of them are novel circuits) by connecting combinations of different designs of these modules. Each of these cells exhibits different power consumption, speed, area, and driving capability figures. Two realistic circuit structures that include adder cells are used for simulation. A library of full-adder cells is developed and presented to the circuit designers to pick the full-adder cell that satisfies their specific applications.

Vesterbacka (1999) explained how exclusive OR and NOR circuits (XOR/XNOR) are used to realize a general full adder circuit based on pass transistors. A six-transistor CMOS XOR circuit that also produces a complementary XNOR output is introduced in the general full adder. The resulting full adder circuit is realized using only 14 MOSFETs, while having full voltage-swing in all circuit nodes. Layouts have been made in a 0.35 μm process for both the proposed full adder circuit and another 16-transistor full adder circuit based on pass transistors. The performance of the proposed full adder is evaluated by comparison of the simulation results obtained from HSPICE for both layouts. The two adders yield similar performance in terms of power consumption, power delay product, and propagation delay. The area is somewhat lower for the proposed adder due to the reduced device count. However, due to two feedback MOSFETs in the proposed adder that need to be ratioed, there is a higher cost in terms of design effort for the proposed adder

Hung Tien Bui et al (2002) discussed that Full adders are important components in applications such as digital signal processors (DSP) architectures and microprocessors. This paper suggest a technique to build a total of 41 new 10-transistor full adders using novel XOR and XNOR gates in combination with existing ones and done over 10,000 HSPICE simulation runs of all the different adders in different input patterns, frequencies, and load capacitances. Almost all those new adders consume less power in high frequencies, while three new adders consistently consume on average 10% less power and have higher speed compared with the previous 10-transistor full adder and the conventional 28-transistor CMOS adder. One drawback of the new adders is the threshold-voltage loss of the pass transistors.

Elguibaly (2000) presented a dependence graph (DG) to visualize and describe a merged multiply-accumulate (MAC) hardware that is based on the modified Booth algorithm (MBA). The carry-save technique is used in the Booth encoder, the Booth multiplier, and the accumulator sections to ensure the fastest possible implementation. The DG applies to any MAC data word size and allows designing multiplier structures that are regular and have minimal delay, sign-bit extensions and data path width. Using the DG, a fast pipelined implementation is proposed, in which an accurate delay model for deep submicron CMOS technology is used. The delay model describes multi-level gate delays, taking into account input ramp and output loading. Based on the delay model, the proposed pipelined parallel MAC design is three times faster than other parallel MAC schemes that are based on the MBA. The speedup resulted from merging accumulate and the multiply operations and the wide use of carry-save techniques.

Goel et al (2006) presented a new design for a 1-b full adder featuring hybrid-CMOS design style. The quest to achieve a good-drivability, noise-robustness, and low-energy operations for deep sub micrometer guided

their research to explore hybrid-CMOS style design. Hybrid-CMOS design style utilizes various CMOS logic style circuits to build new full adders with desired performance. This provides the designer a higher degree of design freedom to target a wide range of applications, thus significantly reducing design efforts. They also classify hybrid-CMOS full adders into three broad categories based upon their structure. Using this categorization, many full-adder designs can be conceived. They will present a new full-adder design belonging to one of the proposed categories. The new full adder is based on a novel xor-xnor circuit that generates xor and xnor full-swing outputs simultaneously. This circuit outperforms its counterparts showing 5%-37% improvement in the power-delay product (PDP). A novel hybrid-CMOS output stage that exploits the simultaneous xor-xnor signals is also proposed. This output stage provides good driving capability enabling cascading of adders without the need of buffer insertion between cascaded stages. There is approximately a 40% reduction in PDP when compared to its best counterpart. During their experimentations, they found out that many of the previously reported adders suffered from the problems of low swing and high noise when operated at low supply voltages. The proposed full adder is energy efficient and outperforms several standard full adders without trading off driving capability and reliability. The new full-adder circuit successfully operates at low voltages with excellent signal integrity and driving capability. To evaluate the performance of the new full adder in a real circuit, they embedded it in a 4- and 8-b, 4-operand carry-save array adder with final carry-propagate adder. The new adder displayed better performance as compared to the standard full adders.

Longa et al (2006) presented a highly area-efficient multiplier-less FIR filter. Distributed Arithmetic (DA) has been used to implement a bit-serial scheme of a general asymmetric version of an FIR filter, taking optimal

advantage of the 4-input LUT-based structure of FPGAs and introduced a modification in the accumulator stage to achieve further savings.

Yoo et al (2005) presented new memory-efficient distributed arithmetic (DA) architecture for high-order FIR filters. The proposed architecture is based on a memory reduction technique for DA look-up-tables (LUTs); it requires fewer transistors for high-order filters than original LUT-based DA, DA-offset binary coding (DA-OBC), and the LUT-less DA-OBC. Recursive iteration of the memory reduction technique significantly increases the maximum number of filter order implementable on an FPGA platform by not only saving transistor counts, but also balancing hardware usage between logic element (LE) and memory.

Mehendale et al (2007) addressed the problem of reducing power dissipation of finite impulse response (FIR) filters implemented on programmable digital signal processors (DSPs) and describe a generic DSP architecture and identify the main sources of power dissipation during FIR filtering. To reduce power dissipated in one or more of these sources even transformations were presented. These transformations complement each other and together operate at algorithmic, architectural, logic and layout levels of design abstraction. Each of the transformations is discussed in detail and the results are presented to highlight its effectiveness and show that the power dissipation can be reduced by more than 40% using these transforms.

Sulistyo & Dong Sam Ha (2003) proposed a Multiplier – Accumulator Unit in which operating frequency would be lower due to wire loads and other parasitic. The simulation was done using SPICE.

Yu-Chi Tsao & Ken Choi (2012) proposed Area Efficient Parallel FIR Filter structure in which hardware savings for symmetric convolutions especially when the length of filter is long.

Ramkumar & Kittur (2012) proposed Low power and Area efficient carry select adder. In this work a simple and efficient gate-level modification significantly reduces the area and power of the CSLA.

Addanki Purna Ramesh proposed radix-2 modified Booth algorithm MAC with SPST gives 7% less power consumption as compared to array MAC.

Rekha K James, in her Ph.D thesis proposed efficient decimal MAC (Multiply Accumulate) architecture for high speed decimal processors based on IEEE 754-2008 Standard for Decimal Floating Point (DFP) Arithmetic.

Shanthala et al proposed VLSI Implementation of Pipelined Multiply Accumulate Unit for Digital signal processing (DSP) applications at 180nm technology.

2.5 CONCLUSION

In this chapter, the study of low power, area efficient and high speed adders and multipliers developed by many researchers have been discussed in the literature. Also, the various techniques for reducing power and increasing efficiency for MAC unit have been presented. In this work the various adders and multipliers are analyzed and the optimized adder and multiplier are selected for designing the MAC unit. Then the Multiplier – Adder combination is simulated and analyzed. From this analysis, for the selected MAC unit, pipelining technique has been introduced for further increasing speed and efficiency. Finally, the proposed MAC unit is implemented in FIR filter.